AGARD-CP-578

AGARD-CP-578

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT
7 RUE ANCELLE, 92200 NEUILLY-SUR-SEINE, FRANCE

AGARD CONFERENCE PROCEEDINGS 578

Progress and Challenges in CFD Methods and Algorithms

(Progrès réalisés et défis en méthodes et algorithmes CFD)

Papers presented and discussions recorded at the 77th Fluid Dynamics Panel Symposium held in Seville, Spain, 2-5 October 1995.

NORTH ATLANTIC TREATY ORGANIZATION

Published April 1996

Distribution and Availability on Back Cover

AGARD-CP-578

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT

7 RUE ANCELLE, 92200 NEUILLY-SUR-SEINE, FRANCE

AGARD CONFERENCE PROCEEDINGS 578

Progress and Challenges in CFD Methods and Algorithms

(Progrès réalisés et défis en méthodes et algorithmes CFD)

Papers presented and discussions recorded at the 77th Fluid Dynamics Panel Symposium held in Seville, Spain, 2-5 October 1995.

North Atlantic Treaty Organization
Organisation du Traité de l'Atlantique Nord

The Mission of AGARD

According to its Charter, the mission of AGARD is to bring together the leading personalities of the NATO nations in the fields of science and technology relating to aerospace for the following purposes:

- Recommending effective ways for the member nations to use their research and development capabilities for the common benefit of the NATO community;
- Providing scientific and technical advice and assistance to the Military Committee in the field of aerospace research and development (with particular regard to its military application);
- Continuously stimulating advances in the aerospace sciences relevant to strengthening the common defence posture;
- Improving the co-operation among member nations in aerospace research and development;
- Exchange of scientific and technical information;
- Providing assistance to member nations for the purpose of increasing their scientific and technical potential;
- Rendering scientific and technical assistance, as requested, to other NATO bodies and to member nations in connection with research and development problems in the aerospace field.

The highest authority within AGARD is the National Delegates Board consisting of officially appointed senior representatives from each member nation. The mission of AGARD is carried out through the Panels which are composed of experts appointed by the National Delegates, the Consultant and Exchange Programme and the Aerospace Applications Studies Programme. The results of AGARD work are reported to the member nations and the NATO Authorities through the AGARD series of publications of which this is one.

Participation in AGARD activities is by invitation only and is normally limited to citizens of the NATO nations.

The content of this publication has been reproduced
directly from material supplied by AGARD or the authors.

Published April 1996

ISBN 92-836-0026-6

Printed by Canada Communication Group
45 Sacré-Cœur Blvd., Hull (Québec), Canada K1A 0S7

Progress and Challenges in CFD Methods and Algorithms

(AGARD CP-578)

Executive Summary

Computational Fluid Dynamics (CFD) now plays an essential role in the design of aerospace vehicles. The ability of numerical methods to accurately simulate complex external and internal aerodynamic flows is crucial to the success of these methods in the design process, and for airplanes leads to improved performance, agility and maneuverability.

In the last decade, considerable progress has been made in the development of numerical methods related to CFD. As a result, various promising CFD schemes and algorithms have been developed. However, they are not currently used in industrial codes. At the same time, new developments in computer hardware and architectures have led to significant advances in parallel computing and multiprocessing. These topics, which are considered likely to constitute pacing items and new challenges in CFD in the near future, formed the framework for the program for this Symposium.

The following subjects were addressed: parallel computing, advanced spatial discretization techniques, unstructured, hybrid and overlapping grids, adaptive meshes, fast implicit and iterative solvers, large eddy and direct numerical simulations of turbulent flows, chemically reacting flows and unsteady aerodynamics. Interesting and new aspects of techniques involving these subjects were discussed, substantiating their extended potential and improved capabilities. Several important directions of research such as aerodynamic shape optimization and multidisciplinary analysis and design were identified, which should be the subject of intensive advanced research in the near future.

The Symposium provided a very valuable opportunity for exchange of information about recent developments and achievements. It can, therefore, be expected to significantly contribute to future important progress in the advancement of numerical techniques used in the design of aerospace vehicles and other flying objects.

Jean-André Essers
Programme Committee Chairman

Progrès réalisés et défis en méthodes et algorithmes CFD

(AGARD CP-578)

Synthèse

L'aérodynamique numérique (CFD) joue désormais un rôle essentiel dans la conception des véhicules aérospatiaux. La capacité des méthodes numériques à simuler avec précision des écoulements aérodynamiques complexes internes et externes est essentielle pour la réussite de ces méthodes dans le processus de conception et pour les aéronefs, elle permet d'améliorer les performances, l'agilité et la manœuvrabilité des appareils.

Au cours de la dernière décennie, des progrès considérables ont été réalisés dans le développement de méthodes numériques se rapportant au CFD. De ce fait, divers algorithmes et diverses méthodes CFD prometteurs ont été développés. Cependant, ils n'ont pas été intégrés aux codes industriels. En même temps, les nouveaux développements en matériel et architectures informatiques ont permis des avancées appréciables dans le domaine du calcul en parallèle et du multitraitement. Ces sujets, qui sont considérés comme susceptibles de constituer les jalons et les nouveaux challenges du CFD dans un avenir proche, ont constitué l'ossature du programme de ce symposium.

Les sujets suivants ont été examinés: le calcul en parallèle, les techniques de discrétisation spatiale avancées, les maillages non-structurés, hybrides et imbriqués, les maillages adaptatifs, les codes de résolution rapides, implicites et itératifs, la simulation des grands tourbillons et la simulation numérique directe d'écoulements turbulents, les écoulements à réaction chimique et l'aérodynamique non permanente.

Des discussions pertinentes ont eu lieu sur des aspects nouveaux et intéressants de techniques se rapportant à ces sujets, confirmant ainsi l'extension de leur potentiel et l'amélioration de leurs capacités. Plusieurs orientations importantes pour la recherche, telles que l'optimisation du profil aérodynamique et l'analyse et la conception multidisciplinaires ont été identifiées comme devant faire l'objet de travaux de recherche avancés intensifs dans un avenir proche.

Le symposium a fourni l'occasion inestimable pour échanger des informations sur les réalisations et les développements récents. Il devrait, par conséquent, représenter une contribution non négligeable aux futurs progrès importants dans l'avancement des techniques numériques pour la conception des véhicules aérospatiaux et d'autres objets volants.

Jean-André Essers
Programme Committee Chairman

Contents

KEYNOTE SESSION
Chairman: J.A. Essers

Recent Publications of the Fluid Dynamics Panel

AGARDOGRAPHS (AG)

Computational Aerodynamics Based on the Euler Equations
AGARD AG-325, September 1994

Scale Effects on Aircraft and Weapon Aerodynamics
AGARD AG-323, July 1994

Design and Testing of High-Performance Parachutes
AGARD AG-319, November 1991

Experimental Techniques in the Field of Low Density Aerodynamics
AGARD AG-318 (E), April 1991

Techniques expérimentales liées à l'aérodynamique à basse densité
AGARD AG-318 (FR), April 1990

A Survey of Measurements and Measuring Techniques in Rapidly Distorted Compressible Turbulent Boundary Layers
AGARD AG-315, May 1989

Reynolds Number Effects in Transonic Flows
AGARD AG-303, December 1988

REPORTS (R)

Parallel Computing in CFD
AGARD R-807, Special Course Notes, October 1995

Optimum Design Methods for Aerodynamics
AGARD R-803, Special Course Notes, November 1994

Missile Aerodynamics
AGARD R-804, Special Course Notes, May 1994

Progress in Transition Modelling
AGARD R-793, Special Course Notes, April 1994

Shock-Wave/Boundary-Layer Interactions in Supersonic and Hypersonic Flows
AGARD R-792, Special Course Notes, August 1993

Unstructured Grid Methods for Advection Dominated Flows
AGARD R-787, Special Course Notes, May 1992

Skin Friction Drag Reduction
AGARD R-786, Special Course Notes, March 1992

Engineering Methods in Aerodynamic Analysis and Design of Aircraft
AGARD R-783, Special Course Notes, January 1992

Aircraft Dynamics at High Angles of Attack: Experiments and Modelling
AGARD R-776, Special Course Notes, March 1991

ADVISORY REPORTS (AR)

Aerodynamics of 3-D Aircraft Afterbodies
AGARD AR-318, Report of WG17, September 1995

A Selection of Experimental Test Cases for the Validation of CFD Codes
AGARD AR-303, Vols. I and II, Report of WG-14, August 1994

Quality Assessment for Wind Tunnel Testing
AGARD AR-304, Report of WG-15, July 1994

Air Intakes of High Speed Vehicles
AGARD AR-270, Report of WG13, September 1991

Appraisal of the Suitability of Turbulence Models in Flow Calculations
AGARD AR-291, Technical Status Review, July 1991

Rotary-Balance Testing for Aircraft Dynamics
AGARD AR-265, Report of WG11, December 1990

Calculation of 3D Separated Turbulent Flows in Boundary Layer Limit
AGARD AR-255, Report of WG10, May 1990

Adaptive Wind Tunnel Walls: Technology and Applications
AGARD AR-269, Report of WG12, April 1990

CONFERENCE PROCEEDINGS (CP)

Aerodynamics of Store Integration and Separation
AGARD CP-570, February 1996

Aerodynamics and Aeroacoustics of Rotorcraft
AGARD CP-552, August 1995

Application of Direct and Large Eddy Simulation of Transition and Turbulence
AGARD CP-551, December 1994

Wall Interference, Support Interference, and Flow Field Measurements
AGARD CP-535, July 1994

Computational and Experimental Assessment of Jets in Cross Flow
AGARD CP-534, November 1993

High-Lift System Aerodynamics
AGARD CP-515, September 1993

Theoretical and Experimental Methods in Hypersonic Flows
AGARD CP-514, April 1993

Aerodynamic Engine/Airframe Integration for High Performance Aircraft and Missiles
AGARD CP-498, September 1992

Effects of Adverse Weather on Aerodynamics
AGARD CP-496, December 1991

Manoeuvring Aerodynamics
AGARD CP-497, November 1991

Vortex Flow Aerodynamics
AGARD CP-494, July 1991

Missile Aerodynamics
AGARD CP-493, October 1990

Aerodynamics of Combat Aircraft Controls and of Ground Effects
AGARD CP-465, April 1990

Computational Methods for Aerodynamic Design (Inverse) and Optimization
AGARD CP-463, March 1990

Applications of Mesh Generation to Complex 3-D Configurations
AGARD CP-464, March 1990

Fluid Dynamics of Three-Dimensional Turbulent Shear Flows and Transition
AGARD CP-438, April 1989

Validation of Computational Fluid Dynamics
AGARD CP-437, December 1988

Aerodynamic Data Accuracy and Quality: Requirements and Capabilities in Wind Tunnel Testing
AGARD CP-429, July 1988

Aerodynamics of Hypersonic Lifting Vehicles
AGARD CP-428, November 1987

Aerodynamic and Related Hydrodynamic Studies Using Water Facilities
AGARD CP-413, June 1987

Applications of Computational Fluid Dynamics in Aeronautics
AGARD CP-412, November 1986

Fluid Dynamics Panel

Chairman: M. C. DUJARRIC
Future Launchers Office
ESA Headquarters
8-10 rue Mario Nikis
75015 Paris — France

Deputy Chairman: Professor C. ÇIRAY
Aeronautical Eng. Department
Middle East Technical Univ.
Inonu Bulvari PK:06531
Ankara — Turkey

PROGRAMME COMMITTEE

Prof. J.A. ESSERS (Chairman)
Université de Liège,
Institut de Mécanique,
Service d'Aérodynamique
rue Ernest Solvay 21
4000 Liège — Belgium

Prof. H. DECONINCK
von Karman Institute for Fluid Dynamics
Chaussée de Waterloo 72
1640 Rhode-Saint-Génese — Belgium

Prof. R.J. KIND
Dept. of Mechanical & Aerospace Eng.
Carleton University
Ottawa, Ontario K1S 5B6 — Canada

Prof. A. BONNET
Dept. Aérodynamique, Ecole Supérieure
de l'Aéronautique et de l'Espace
10 Avenue Edouard Belin, BP 4032
31055 Toulouse Cedex — France

ICA O.P. JACQUOTTE
Direction de la Recherche et de la
Technologie — DRET/STRDT/G6
4 Avenue de la Porte d'Issy
00460 Armées — France

M. R. LACAU
Aérospatiale Missiles (E/ECN)
Centre des Gâtines
91370 Verrieres le Buisson — France

Dr. Ing. H. KORNER
Direktor — Institut für Entwurfsaerodynamik
der DLR
Lilienthalplatz 7
D-38108 Braunschweig — Germany

Dr. A.G. PANARAS
P.O. Box 64053
Athens 157 10 — Greece

Mr. M. BORSI
ALENIA Aeronautica
Corso Marche 41
10146 Torino — Italy

Prof. Ir. J.W. SLOOFF
National Aerospace Laboratory NLR
Anthony Fokkerweg 2
1059 CM Amsterdam — Netherlands

Prof. Dr. T. YTREHUS
Division of Applied Mechanics
The University of Trondheim
The Norwegian Inst. of Technology
N-7034 Trondheim — NTH — Norway

Prof. A.F. de O. FALCAO
Depart. Engenharia Mecanica
Instituto Superior Tecnico
1096 Lisboa Codex — Portugal

Dr. R. CORRAL
Departamento de Mecanica de Fluidos
Industria de Turbopropulsores (ITP)
Carretera Torrejon Ajalvir, Km 3.5
28850 Torrejon de Ardoz (Madrid) — Spain

Prof. J. JIMENEZ
Escuela Tecnica Superior de Ingenieros Aeronauticos
Departamento de Mecanica de Fluidos
Plaza del Cardenal Cisneros 3
28040 Madrid — Spain

Prof. Dr. U. KAYNAK
TUSAS — Havacilik ve Uzay San. A.S.
P.K. 18 Kavaklidere, 06690
Ankara — Turkey

Prof. D.I.A. POLL
Head of the College of Aeronautics
Cranfield University — Cranfield
Bedford MK43 OAL — U.K.

Prof. B. CANTWELL
Stanford University
Dept. of Aeronautics & Astronautics
Stanford, CA 93405 — U.S.A.

Dr. L.P. PURTELL
Mechanics & Energy Conversion Division
Code 333 — Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5660 — U.S.A.

PANEL EXECUTIVE

Mr. J.K. MOLLOY

Mail from Europe
AGARD-OTAN
Attn: FDP Executive
7, rue Ancelle
92200 Neuilly-sur-Seine
France

Mail from USA and Canada
AGARD-NATO
Attn: FDP Executive
Unit PSC 116
APO AE 09777

Tel: 33 (1) 4738 5775

Technical Evaluation Report
AGARD Fluid Dynamics Panel Symposium on
"Progress and Challenges in CFD Methods and Algorithms“

N. Kroll

Institute of Design Aerodynamics
DLR, 38108 Braunschweig
Lilienthalplatz 7, Germany

SUMMARY

The Fluid Dynamics Panel of AGARD conducted a Symposium on "Progress and Challenges in CFD Methods and Algorithms“ in Seville, Spain, on October 2-5, 1995. The purpose of this symposium was to identify and discuss topics which are likely to constitute pacing items and challenges in Computational Fluid Dynamics. Sessions were devoted specifically to parallel computing, advanced discretization schemes and advanced grid structures. Topics also include adaptive meshes, fast iterative methods and algorithmic aspects for the computation of reacting flows and unsteady flows. In this evaluation report an attempt is made to point out the critical issues for each particular subject and to assess how far they were addressed by the conference papers. Some general concluding remarks and recommendations are given.

1. INTRODUCTION

The 77th Meeting of the AGARD Fluid Dynamics Panel was held from the 2nd to the 5th of October, 1995, in Seville, Spain. The symposium was focused on "Progress and Challenges in CFD Methods and Algorithms“. The background and need for such a meeting was stated in the call for papers:

> "The design of aerospace vehicles strongly depends on the ability of numerical methods to simulate complex flow fields. In the last decade, considerable progress has been made in the development of numerical methods related to CFD. As a result, various promising CFD schemes and algorithms have been developed which are not yet currently used in industrial codes. At the same time, new developments in computer hardware and architectures have led to significant advances in parallel computing and multiprocessing.“

It must also be stated that despite the recent advances CFD still suffers from deficiencies in accuracy, robustness and efficiency for complex applications, such as complete aircraft flow predictions. From the aeronautical industry‘s point of view, CFD is expected to deliver:

- detailed viscous flow analysis for complex geometries at realistic Reynolds numbers
- accurate prediction of aerodynamic data
- fast response time per flow case at acceptable total costs
- aerodynamic optimization of aircraft components/complete aircraft
- interdisciplinary analysis of aircraft (aerodynamics + structure + flight control)

In order to meet these requirements, improvements in CFD are needed in all areas.

Based on this, the aim and scope of the symposium were set by the program committee in the call for papers as follows:

> "The symposium will focus on those topics which are likely to constitute pacing items and new challenges in CFD. Its aim is to bring together scientists and engineers working on new numerical developments in different fields of interest to the aerospace sciences and industrial communities.
>
> Papers may address a broad range of research fields of current interest. A list of possible topics includes (but is not limited to) the following:
>
> - Unstructured grid, hybrid, adaptive, multiblock and grid embedding methods and algorithms
> - Implicit and iterative methods for Euler and Navier-Stokes equations, fast iterative solvers (multi-grid, Krylov subspace techniques)
> - Numerical techniques for parallel computing and multiprocessing

- Advances in accurate capturing techniques for shock waves and contact discontinuities, TVD high resolution schemes, multidimensional upwinding
- Numerical algorithms and problems specifically related to the implementation of turbulence models and to the simulation of nonequilibrium chemically reacting flows
- Numerical accuracy assessment.

In order to limit the scope of the symposium, papers essentially devoted to grid generation techniques and turbulence or chemistry modelling are not encouraged.“

The symposium spanned three and one-half days and the program listed 38 technical papers coming from 13 countries, of which 36 papers were presented. The program committee organized a keynote session, eight major sessions and a general discussion at the end of the meeting. The following table presents the topics covered by the symposium. Although many papers addressed several topics, they are categorized in this table based on their central focus. The papers are listed in chapter 4 in the order of their presentation.

topic	reference
invited papers	[1], [2], [3]
parallel computing	[4], [5], [6], [7], [21], [25], [36]
advanced spatial discretization schemes	[9], [10], [12], [14], [15], [16], [17]
unstructured grids, hybrid grids, overlapping grids, meshless techniques	[8], [11], , [28], [29]
adaptive schemes	[13], [20], [24], [34]
fast implicit and iterative solvers	[18], [19], [26]
turbulent flows, LES / DNS	[22], [23]
chemically reacting flows	[30], [31]
unsteady flows	[[27], 32], [35], [33]

It is worthwhile to note that the majority of the papers were concerned with parallelization of CFD methods, use of more flexible grid structures and development of advanced discretization schemes including adaptive methods. This may reflect the contemporary trends of CFD research in most aeronautical companies, goverment research laboratories and universities. Surprisingly, except for the keynote paper by Jameson, no technical paper addressed optimization and interdisciplinary analysis which, in the author‘s opinion, are major challenges in CFD.

The evaluation undertaken in this report attempts to cover two aspects. Chapeter 2 comprises summaries of the presented papers for each topic given in table 1. It is not intended to give an extensive review of all individual papers, but instead, for each particular subject it is aimed to identify the critical issues and to assess how far they were addressed by the papers. In chapter 3, concluding remarks are presented indicating to what degree the aims of the meeting and the needs of the aerospace community were met. Furthermore, recommendations arising from the meeting are given.

2. SYNOPSIS OF THE PAPERS

With respect to the theme of the meeting "Progress and Challenges in CFD Methods and Algorithms", in the evaluator‘s opinion, many papers of high quality were given, which represent the current status of CFD, focus on unresolved issues and present new important directions of development to overcome current deficiencies. On the other hand, many papers of lower quality were presented. Some of them did not meet the main focus of the symposium, several others did not reflect the present status of CFD or were largely redoing or reinventing well established topics that have been known in the literature for some time.

2.1 Invited Papers

Keynote papers were provided by *A. Jameson*, *P. Rubbert* and *D. Knight.*

Jameson [1] gave an excellent overview of present status, challenges and future developments in computational fluid dynamics. He addressed the essential requirements on numerical simulation for their effective industrial use. Assured accuracy, acceptable computational and human costs as well as fast turn around were identified as major issues. In his opinion, more sophisticated algorithms are required in order to substanially reduce computational costs. Improved methods should include higher order schemes, advanced acceleration methods, fast inversion methods for implicit schemes and the effective exploitation of massively parallel computers. The paper reviewed modern numerical methods and addressed several issues in algorithm design. In particular, a unified approach to design accurate and efficient shock capturing algorithms was presented. Some examples of state-of-the-art calculations, which can be performed in an industrial environment, were given. *Jameson* pointed out that beside the transition to more sophisticated algorithms, the present challenge is to extend the effective use of CFD techniques to more complex applications. As key problems, he identified turbulent flows at Reynolds numbers associated with full scale flight, chemically reacting flows, combustion and unsteady flows. Furthermore, multidisciplinary analysis, aerodynamic shape optimization and in the long run multidisciplinary optimization were designated as important future target areas of CFD. In his presentation, *Jameson* outlined a very promising technique for efficient three-dimensional shape optimization based on control theory. He demonstrated a successful design of a swept wing with very low wave drag within 40 design iterations. In this example, the flow was modeled by the Euler equations. He mentioned that with this technique, even in the case of three-dimensional flows, the computational requirements are so moderate that the calculations can be performed with workstations such as the IBM RISC 6000 series.

In summary, the invited paper delivered by *Jameson* gave a precise outline of the scope of the symposium and the expected outcome of the meeting.

In his presentation [2], *Rubbert* focused his remarks on challenges and pacing items in CFD that extend beyond the technical ones. He pointed out that the key to developing better airplanes or better CFD is the same, namely to analyze, understand and improve the processes by which airplanes or CFD are created. *Rubbert* called the process by which CFD capabilities are created the research engine. Such a research engine involves industry, academia and government, and the three components interact with each other as a system. In the past this system functioned quite well, but, in his opinion, it has been almost disconnected from the customers of CFD research, namely the practicing design engineers. Impressive results of research have been achieved, but they were not necessarily applicable by industry. The paper pointed out many principal characteristics and attributes which an improved, properly functioning research engine should have. The leading principles are customer focus and customer satisfaction. Two further key factors were identified which will pace the change of the research engine. The first is a two-way, more intensive communication between the research community and the engineering community in industry. The second is a modification of the evaluation system of the research work towards more industrial applicability. As stated by *Rubbert*, this is the responsibility of the money givers who inhabit the research engine.

In summary, *Rubbert*'s paper performed a general critical assessment of today's system of research and its stage of change. His observations represent the pragmatic point of view of industry, from which the interest of researcher's basic scientific findings are less emphasized. This paper makes CFD researchers sensitive to industrial needs, but some specific views of aeronautical industry on the status of CFD and future requirements would have been desirable.

The paper by *Knight* [3] presented an overview of parallel computing in computational fluid dynamics. In the first part of the paper the basics of parallel computing were addressed, including the introduction of the distinct levels of parallelism, the classification of parallel computer architectures and the description of the two basic programming paradigms, namely message passing and data parallelism. The second part focused on several key issues in the context of code development for parallel computing. Dynamic load balancing and scalability were identified as critical issues for complex CFD applications carried out on massively parallel computers. Furthermore, a major concern of parallel computing is portability. Here, *Knight* discussed current research activities, including the development of message passing standards (e.g. PVM, MPI) and data parallel programming language standards (e.g. HPF). In his presentation, *Knight* pointed out that in the U.S. aerospace industry has taken a leading role in the application of parallel computing to practical analysis and design. In the past few years several major aerospace corporations have developed extensive networks of workstations for routine applications. Several examples were given in the paper.

Knight's presentation provided a basic introduction into the field of parallel computing. The fundamental terminology were explained and all critical issues were discussed. Therefore, the paper was very helpful for the understanding and assessment of the following technical papers which dealt with parallelization. Unfortunately, the paper did not discuss the potentials and limitations of high performance parallel computers to tackle large scale applications or new challenges in CFD. Results were only presented for networks of loosely coupled workstations.

Although papers on grid generation techniques were explicitly not encouraged by the call for papers, an invited paper on status and progress of both structured and unstructured grid generation for complex configurations might have been desirable. The turn around time and accuracy of the numerical simulation of industrial applications very often depend on the capability of the available grid generation procedure. Therefore, for the critical assessment of numerical algorithms using structured, unstructured or hybrid meshes, the capabilities and limits of the underlying grid generation techniques have to be taken into account.

2.2 Parallel Computing

Parallel computing is an important means to cut down turn around time and computational costs of large scale applications. Furthermore, it is believed that the exploitation of massively parallel computing is the key to tackle new grand challenges in CFD such as multidisciplinary analysis and optimization. In the last several years a wide variety of parallel architectures have become available which differ in the design of the CPU's (vector versus RISC processor), the memory organization (e.g. shared versus distributed memory) and the communication system (hardware and software). For the future some of the vendors promise substantial increase of computational power in both memory and CPU. One of the main issues in parallel computing is the design of numerical algorithms which efficiently exploit the capabilities of the parallel hardware. Especially in the case of distributed memory machines, this is a non trivial task. The important aspects in designing parallel algorithms for these architectures are partioning of data and computation among the processors, communication at the internal boundaries, load balancing and overhead due to communication and extra computations. Simpler algorithms, such as explicit schemes, parallelize quite easily and they lead to high performance on most parallel computers. However, due to their poor convergence rates they are overall much less efficient than implicit schemes, although the latter ones generally perform far below the peak of the parallel machines due to the more intensive and more global communication involved. The adjustment and further development of more sophisticated algorithms such as multigrid and domain decomposition methods on parallel architectures are very promising. In contrast to explicit schemes, they provide global distribution of information, however in a much more efficient way than traditional implicit schemes. Further research in this direction is needed in order to efficiently exploit the capabilities of parallel computers.

In this symposium, papers [4], [5], [6], [7], [21], [25] and [36] dealt mainly with parallel computing and covered various aspects thereof. The paper by *Eisfeld et al.* [4] stressed the issue of portability. They described the portable parallelization of a state-of-the-art block-structured multigrid solver for industrial CFD applications. Portability is achieved through the use of a message passing based high level communication library. This library supports any operation which is necessary in parallel mode and involves communication between different processes. Performance measurements on a large variety of computers of different architectures demonstrated the comprehensive portability of the code. Applications included inviscid computations for a generic aircraft consisting of wing/body/pylon/engine and viscous calculations for a wing-body configuration on a computational mesh with 6.6 million points. The paper showed that the complexity of today's problems in applied aerodynamics can be tackled with parallel computers. It also revealed the necessity for an automatic and effective load balancing tool that allows the mapping of an initial block structure to a higher number of processors than given blocks. Details on the parallel efficiency of the multigrid method used in the applications were not given.

The papers by *Wissink et al.* [6], *Dias d'Almeida et al.* [7] and *Badcock et al.* [36] focused on the parallel implementation of implicit Euler/Navier-Stokes solvers. In [6] for example, two modifications of the well known implicit LU-SGS scheme (Lower-Upper Symmetric Gauss-Seidel) were presented. The first replaces the Gauss-Seidel sweep in LU-SGS with a Jacobi-like sweep which only requires nearest neighbor communication and is therefore easy to parallelize. The second one is a hybrid approach that couples the global Jacobi type communication with the more efficient Gauss-Seidel sweep on each subdomain. In both strategies multiple sweeps are required in each subdomain in order to maintain the convergence behavior of the baseline LU-SGS method. Both strategies have been investigated in detail with respect to parallel speed-up, convergence rate and computational efficiency. Inviscid calculations for 3-D hovering helicopter blades demonstrated that the hybrid strategy is a promising implicit scheme for parallel computers with a smaller number of powerful processors.

The presentations delivered by *Pinelli et al.* [5] and *Streng et al.* [21] addressed the parallelization of algorithms for DNS and LES. In [21] the various aspects of the parallel implementation of a typical higher order DNS solver based on domain decomposition were discussed. The intrinsic or algorithmic efficiency has been defined (see also [4]), which deals with the parallelizability of a given algorithm, regardless of the machine. Based on some analysis, the authors showed that due to extra floating point operations at inner block boundaries the algorithmic efficiency decreases rapidly as the spatial discretization increases, that is, as the corresponding stencil-size grows. Test calculations on different parallel architectures indicated that the machine efficiency is even considerably lower than the algorithmic efficiency. Furthermore, the paper reported on first experience that has been gained for the implementation of the DNS solver on a SGI Power Challenge Array (4 nodes each comprised of a 16-CPU shared memory parallel machine) using a combination of fine-grained (shared memory) and coarse-grained parallelism (explicit message passing). The results were very promising, however, this parallelization strategy needs further investigation.

The paper by *Sibilla and Vitaletti* [25] did not show any parallel computations, but it addressed several important aspects of multiblock-structured grid algorithms in a parallel computing environment. As in [4] the management of data communication between adjacent blocks is provided by a parallel library (PARAGRID) which ensures that the same average values are assigned to all replicas of the same boundary node owned by different blocks. The paper discussed the influence of block subdivision on accuracy and efficiency within the framework of a multigrid scheme. The solution algorithm has been modified in order to account for the presence of locally unstructured topologies at block boundaries (singular points). For some test cases it could be demonstrated that the convergence of the numerical method could only be ensured with this modification.

In conclusion, most of the papers focused on some specific algorithmic aspects of parallel computing. Effort was essentially put in adjusting sequential algorithms rather than developing new parallel schemes. Only a few large scale CFD applications have been presented demonstrating the capabilities and limits of parallel architectures for industrial CFD applications. One of the main challenges for parallel complex applications is the load balanced partioning of the flow domain, which is essential for obtaining optimal machine performance. This important issues were hardly addressed in the conference.

2.3 Advanced Spatial Discretization Schemes

Although in the last decade extensive research has been ongoing towards the development of accurate Euler and Navier-Stokes solvers, the improvement of spatial discretization schemes is still a major concern in CFD. Suitable discretization schemes are expected to offer certain properties. These are conservation, at least second order accuracy in smooth flow regions and sharp resolution of discontinuities and viscous shear layers. High resolution of all physical phenomena is required on a computational mesh with a minimum number of grid points. Furthermore, the spatial discretization should support a robust and efficient time integration. Recently, substantial progress has been made in this area and many different promising approaches for the improved discretization of the Euler and Navier-Stokes equations are known in the literature. Among these are e.g. improved shock capturing algorithms based on flux difference and flux vector splitting, multidimensional upwinding, residual distribution schemes and kinetic flux splitting. These methods have been investigated in detail for one and two-dimensional flows . Very often, however, their superiority to conventional methods have only been demonstrated for simple test cases. Therefore, the key issue remains the manifestation of the improved abilities of the advanced methods for relevant 2-D and 3-D viscous flows around more complex geometries.

At the symposium several papers [9], [10], [12], [14], [15], [16] and [17], were specifically devoted to improvements of the spatial discretization of Euler/Navier-Stokes solvers. The paper by *Delanaye et al.* [9] presented the development of a new quadratic reconstruction finite volume scheme for unstructured polygonal meshes. The most frequently employed linear cell reconstruction of the flow variables requires sufficiently regular grids for second order accuracy and it results in a first order scheme for irregular meshes. In contrast to this, the proposed quadratic reconstruction provides a full second order scheme even for very irregular meshes. In order to avoid spurious oscillations in the vicinity of discontinuities, the quadratic reconstruction is switched to a monotone constant one with the help of a properly defined discontinuity detector. The method is designed to deal with adaptive unstructured grids consisting of cells with an arbitrary number of edges. Time integration is performed by an implicit scheme based on Newton-Krylov techniques. The efficiency and high accuracy of the numerical method were demonstrated for various 2-D inviscid and viscous laminar computations including test cases with locally distorted meshes. However, the method needs to be carefully investigated for 3-D complex geometries and turbulent flows at high Reynolds numbers where highly irregular meshes are expected. Furthermore, the sensitivity of the quadratic reconstruction with respect to implementation of wall boundary conditions has to be investigated.

The paper delivered by *Villedieu et al.* [10] presented a second order scheme based on kinetic flux splitting. The main feature of this approach is that under a CFL like condition density and energy can be proved to remain nonnegative. This makes the method very attractive for predictions of flow fields with near vacuum conditions, such as flows around hypersonic vehicles. Promising results were shown for 2-D supersonic and hypersonic inviscid flows in comparison with the classical Roe flux difference split scheme. First 3-D results for a wing alone application were presented which do not yet allow the assessment of the approach for 3-D more complex application. Furthermore, detailed remarks on the convergence behavior of the method were missing.

The paper by *Briggs et al.* [13] addressed the effect of certain parameters of a classical TVD scheme on the solution of the specific viscous flow problem of a transverse jet interacting with a supersonic flow. As already known from many other applications, the size of the entropy correction parameter and the choice of the flux limiter can significantly influence the accuracy of the viscous solution, especially on coarse meshes or meshes with improper point distribution in the boundary layer.

The papers [14], [15] and [16] were devoted to multidimensional upwinding. *Vinckier et al.* [14] presented a so-called flux filter scheme which operates on the discrete cell flux balance and assigns filtered parts of the residuals to the corresponding cell vertices according to the characteristic propagation directions. For stability reasons some kind of artificial viscosity, similar to the classical central scheme, has to be added, which is an unwelcome feature in the framework of a multidimensional upwind approach. Results for 2 D inviscid and viscous laminar flows on both structured and unstructured meshes were presented indicating high resolution of flow features like shocks and expansion fans. However, for airfoil flows improved accuracy of the method compared to classical central or upwind schemes was not demonstrated.

The paper by *Paillère et al.* [15] reviewed recent developments in multidimensional upwind schemes based on the residual decomposition or fluctuation splitting approach. Substantial progress has been made in the implementation of truly multidimensional upwinding in which unlike the standard upwind schemes the upwind biasing is determined by properties of the physics rather than the computational mesh. For scalar conservation laws various advection schemes distributing the conservative flux balance to only the downstream nodes have been developed. These schemes can be designed such that properties as conservation, positivity and second order accuracy are guaranteed. It was reported that the extension of these schemes to Euler/Navier-Stokes equations is straight forward provided that a conservative linearization can be found. This can easily be achieved for triangular meshes, whereas for quadrilateral meshes it is more difficult and still subject of ongoing research. The paper presented various numerical examples for 2-D flows demonstrating the ability of the residual decomposition approach. In particular, the results indicate the improved resolution of flow discontinuities which are not aligned with mesh lines. Unfortunately, the issue of accurate prediction of turbulent viscous flows was not addressed in the paper. Furthermore, no 3-D results were shown. The residual decomposition schemes have been successfully combined with implicit methods and solution adaptive techniques.

The paper by *Van Ransbeeck and Hirsch* [16] presented an alternative approach for multidimensional upwind schemes on structured meshes. In this framework the numerical flux is formulated using the artificial dissipation concept. The diffusive contribution is constructed with directional terms, whereas the antidiffusive term is designed according to the direction of the convection speed and to variations of the solution in different mesh directions. The paper presented a classification of first and second order accurate schemes that have respectively minimum and zero cross diffusion. Second order monotone schemes have been developed using the concept of non-linear limiter functions applied to multidimensional ratios of flux differences. Extensions of the scalar dissipation model to the Euler/Navier-Stokes equations have been achieved through a characteristic decomposition. Different choices for the propagation direction are possible. Promising results were presented for 2-D and 3-D supersonic test cases showing comparable or somewhat improved accuracy with respect to classical second order dimensional-split upwind schemes. However, no results for subsonic test cases such as flow past an airfoil were shown. Therefore, a comprehensive assessment of the concept is not possible at the current stage of research.

In summary, promising approaches were presented, which aimed at improving the accuracy of state-of-the-art Euler/Navier-Stokes solvers. In particular, the higher order reconstruction approach and the multidimensional upwind schemes offer properties which in theory make them superior to standard algorithms. Numerical results for various two dimensional test problems support this. However, in order to push the implementation of these advanced techniques into 3-D production codes for viscous flow calculations, further investigations are required. A critical assessment should include sensitivity studies with respect to grid fineness and grid regularity for transonic 2-D and 3-D viscous flows. It should be clarified whether with these new concepts substantial progress can be made towards accurate drag prediction for 2-D and 3-D configurations at relevant Reynolds numbers.

2.4 Unstructured Grids, Hybrid Grids, Overlapping Grids and Meshless Techniques

The key problem of numerical simulation of complex configurations is the construction of an appropriate grid to represent the computational domain of interest. Grid generation is the decisive factor concerning the turn around time of simulations for industrial applications. Essentially two alternative strategies exist, namely structured and unstructured meshes. Currently, block-structured body-fitted meshes are most widely used. They have been proved to be well suited for viscous calculations and they form the building blocks of most of the industrial state-of-the-art production codes. However, with this approach, grid generation for complex geometries itself is the major challenge. Various strategies are being developed to simplify the grid generation problems. Among these are the overlapping grid techniques where the structured grids of various blocks may overlap. The alternative approach is to divide the computational domain into an unstructured assembly of computational cells by using tetrahedra or general polygonal volumes. In contrast to structured meshes, this strategy substantially simplifies the discretization of complex geometries. On the other hand, it complicates the design of accurate and efficient algorithm. While in the past promising and flexible unstructured methodologies have been developed for inviscid flows, the accurate calculation of viscous flows using unstructured meshes is still an important issue of current research. In particular, efficient simulation of high Reynolds number flows requires extremely stretched cells, which in the case of tetrahedral meshes lead to tetrahedra with acute angles. This may cause numerical errors, at least for classical schemes currently used in industrial codes. An interesting alternative is the use of hybrid grids consisting of tetrahedra and hexahedral or prismatic cells. It offers the possibility of combining the flexibility of tetrahedral meshes with the accuracy of regular grids in the boundary layer. The ability of this approach to simulate turbulent flows around complex 3-D geometries is still to be verified.

Some of the before mentioned issues concerning the use of more flexible grid structures were addressed during the meeting. The paper delivered by *Ramakrishnan et al.* [8] presented the experience on unstructured grid computations gained at Rockwell Science Center over the past several years. One of the most important lessons they have learned from many 3-D applications is the fact that in spite of all the advances that have been made in the field of unstructured procedures, on comparable grid fineness structured-grid simulations yield more accurate solutions. The authors concluded that for inviscid flows unstructured Euler solvers have a clear edge over their structured counterparts. This is due to the fact that the solution of Euler equations, unlike Navier-Stokes equations, does not require very fine meshes in the vicinity of solid bodies. Therefore, unstructured grid generation becomes much easier to handle and several computations for many different configurations can be carried out in a matter of a few weeks. In the case of viscous flows, however, the stringent resolution requirements in the wall normal direction makes structured solvers more suitable for efficient calculations. In the framework of unstructured meshes, paper [8] presented a generalization of the implementation of boundary conditions which allows the specification of interior boundaries anywhere in the computational domain. This concept allows the effective computation of moving bodies, like in the case of aircraft store release. However, no detailed numerical results were shown.

The paper by *Galle* [28] addressed the solution of Euler and Navier-Stokes equations on hybrid grids consisting of prismatic cells near the body surface and tetrahedral cells elsewhere. The use of prismatic cells offers the possibility to efficiently and accurately resolve regions such as boundary layers by applying high aspect ratio cells in the respective areas. An upwind finite volume scheme has been implemented on an auxiliary mesh of control volumes. This dual mesh formulation guarantees conservation in the entire flow field and in particular at interfaces between prismatic and tetrahedral domains. The integration in time is performed by an explicit multistage scheme accelerated by a multigrid technique based on agglomeration of control volumes. Promising numerical results were shown for 3-D inviscid and 2-D viscous flows demonstrating the ability of the method. However, further 3-D viscous calculations for more complex geometries are required to proof the concept of the hybrid mesh approach.

The paper delivered by *Brenner* [29] presented a computational procedure to simulate rocket stage separation. The Euler equations with mixing gases are solved with an upwind finite volume method on unstructured meshes, which may consist of a combination of tetrahedral, prismatic and hexahedral cells. In order to simulate the motion of bodies, a conservative overlapping grid technique have been implemented. A temporal adaptive algorithm is used to calculate the unsteady flow field. A very impressive application was shown, which however did not allow a critical assessment of the method concerning its accuracy.

The interesting concept of meshless simulation techniques for fluid flow problems was presented by *Onate et al.* [11]. According to the work of Batina the discrete approximation of the governing equations uses a cloud of arbitrary points. Unlike conventional meshes, no fixed connectivities between the points is needed and therefore grid properties like regular cells or non negative cell volumes are not required. A weighted least square interpolation is used to construct a linear or quadratic function from the values given at the arbitrary points in the local interpolating domain. First examples for the solution of the 1-D convection diffusion equation and for 2-D compressible inviscid flows were shown. For these calculations the points generated by an unstructured mesh have been used. It was pointed out that major difficulties of this approach are the definition of the local interpolating domain and the selection of the most significant points for the interpolation in each domain. The interpolation strategy strongly influences the quality of the solution. Another drawback is that the method is not conservative. Furthermore, it is quite difficult to access the accuracy of the numerical procedure if a set of arbitrary points is used. However, since in theory the meshless approach does not require a suitable grid of high quality and allows an efficient adaption strategy, further research in this area is encouraged.

The papers reviewed above addressed different and in comparison to structured meshes more sophisticated grid strategies which are expected to improve or even enable the simulation of 3-D complex configurations. Promising results for various, mostly inviscid test cases were shown. However, the abilities of the advanced techniques to accurately calculate turbulent viscous flows around more complex geometries were not demonstrated at the conference.

2.5 Adaptive Schemes

In recent years adaptive grid methods for computational fluid dynamics have gained popularity due to their potential to provide highly accurate solutions on the basis of cost-effective calculations. In contrast to global reduction of the mesh interval, very fine mesh cells are restricted to those regions where flow features need high grid resolution; elsewhere the computational grid may be quite coarse. Grid adaption methods can be categorized into either point redistribution or mesh embedding/enrichment. Point redistribution schemes maintain a constant number of points, which are moved such that they congregate near flow features. This technique can be easily implemented into existing structured and unstructured flow solvers. However, it can lead to quite skewed grids, especially in the case of structured meshes. The grid embedding technique add points to the existing grid. This procedure maintains the global grid accuracy outside embedded regions and simultaneously increases the accuracy in the embedded regions. The key issue for adaptive methods is the design of suitable error estimators. By far the most common approach is to use physical features such as local solution gradients. These indicators efficiently detect high-gradient regions such as shock waves, however the global error may not necessarily be reduced and the numerical solution may depend on the adaption pattern. Recently more advanced, direct error estimators are used. They are either based on the discretization error, which may be estimated by comparing quantities calculated on two different fine meshes, or on the residual error. This strategy is very promising but needs further research, especially if it is applied to viscous flows. In conclusion, adaptive strategies are considered as one of the pacing items of algorithmic research. Issues which have to be clarified for complex applications are the development of suitable adaption criteria allowing grid independent solutions and dynamic load balancing for parallel computing.

The issue of adaptivity has been addressed by many conference papers. Papers [9], [15] and [34] reported on adaptive refinement in the context of unstructured solvers based on insertion and removal of grid points. The papers [15] and [34] presented an adaption strategy which relies on a finite element error estimator. Whereas in [15] the application is restricted to steady 2-D inviscid flows, *Friedrich et al.* [34] presented a dynamical adaption for various 2-D unsteady flows. The error indicator, which has been proved reliable for many inviscid calculations, is currently being extended to the Navier-Stokes equations. First grid adaptions for viscous flows were shown. Furthermore, the finite element residual has been successfully used for a 3-D inviscid wing application. The adaptive unstructured solver of [34] has been parallelized on the basis of an intelligent dynamic load balancing procedure for performance controlled domain decomposition. In the parallel mode an explicit time integration is employed, whereas on a sequential computer unsteady calculations are carried out by an implicit method using a preconditioned GMRES technique.

The paper by *Becker et al.* [24] addressed the adaptive grid refinement for block structured solvers. In this concept, locally refined mesh blocks are patched into the existing mesh. The additional fine subblocks are connected with the original mesh via the multigrid technique. The level of local truncation error is used as error indicator. Following the idea of Brandt, truncation error estimates can be extracted directly from the multigrid cycle. So far, the refinement procedure is set up outside the flow solver. First results presented for 2-D and 3-D inviscid and viscous test cases show the feasibility of the strategy of subblock refinement. However, considerable more work is required to establish a fully automatic, robust and accurate adaption method.

Van der Vegt et al. [20] presented a hexahedron based grid adaption procedure. The method uses the discontinues Galerkin finite volume formulation with local grid enrichment. A directional grid adaption is employed which allows subdividing of cells, independently in each of the three local grid directions. This anisentropic grid refinement is expected to be more efficient in capturing local flow phenomena than isentropic refinement, since many flow features are one-dimensional. The sensor uses primitive variables and is constructed such that it prevents regions with discontinuities from constantly dominating the local grid refinement procedure. The capability of the adaptive method was demonstrated by calculations of the inviscid transonic flow around a generic delta wing. From the author‘s viewpoint, the hexahedral based adaptive solver is a good candidate for large eddy simulations (LES), because it offers the opportunity to accurately capture viscous sublayers with successively fine grids through local grid refinement. LES results, however, were not shown.

Grid adaptive procedures based on point redistribution were discussed in the papers [12], [27], [33]. This technique was mainly used in the framework of moving grids for unsteady calculations.

In conclusion, various adaptive strategies were presented. The important issue of developing a suitable indicator for adaption was addressed. Various error estimators have been proposed and successfully applied to inviscid flows. However, further research is needed to establish efficient and robust adaptive methods for viscous flows.

2.6 Fast Implicit and Iterative Solvers

As numerical flow simulations pave their way into the practical aerodynamic design process, the need for efficient algorithms to solve the spatial discretized Euler/Navier-Stokes equations has become very obvious. Many solvers still used in current aerospace development programs exhibit slow convergence towards the desired steady state solutions which leads to high computer costs and long turn around times. Consequently, there is a substantial amount of research work focused on methods for convergence acceleration. Promising approaches are the multigrid time-stepping technique and the Newton iteration with fast iterative solvers. In structured codes multigrid techniques based on explicit multistage schemes are widely used and they have been proved to yield good convergence rates for many practical applications. However, for the numerical simulation of high Reynolds number flows, the convergence of the standard multigrid schemes considerably slows down. This is due to the stiffness of the numerical problem, which is introduced through the high-aspect ratio cells required for the efficient solution of such flow fields. Therefore, one of the key issues concerning algorithmic development is the design of appropriate multigrid components, such as smoothing and grid transfer operators, which efficiently tackle high aspect ratio cells.

Interest in fast iterative methods has been mainly motivated by unstructured solvers. It was shown that coupling Newton‘s method with iterative solvers for the inner iteration is an effective approach for solving the large systems of nonlinear equations arising from the discretization of Euler and Navier-Stokes equations. An interesting feature of Newton‘s method is its ability to provide superlinear asymptotic convergence. On the other hand, efficient iterative schemes based on Newton‘s iteration require excessive memory allocations for three dimensional applications. Therefore, strategies have to be developed which eliminate the large storage requirements but still remain the favorable convergence characteristics of Newton‘s method.

The paper by *Pulliam et al.* [19] gave an excellent overview of the potentialities and drawbacks of Newton‘s method applied to CFD solvers. For practical reasons, in each Newton iteration the large block banded matrix is solved by an iterative matrix solution method. In particular, the paper addressed the class of Krylov subspace methods known as GMRES. It presented practical aspects and implementation issues of these methods. The main components of the Newton-GMRES approach, such as evaluation of the Jacobian, matrix-vector multiply and matrix preconditioning, were discussed with respect to global convergence behavior, memory requirements and accuracy. Trade-offs between full Newton and approximate Newton and other pertinent approximations were investigated. The Newton-GMRES solver was analyzed in the framework of a structured and unstructured 2-D Navier-Stokes code. In both cases very promising results were shown. Calculations with similar methods were also carried out in papers [9], [15]. It can be concluded that optimal strategies which ensure favorable convergence characteristics will lead to excessive memory requirements. No 3-D calculation with Newton-Krylov subspace techniques were presented at the conference.

The paper by *Cambier et al.* [26] proposed a new implicit algorithm called DDLU factorization. Compared to the classical ADI factorization, this strategy enables a reduction in both CPU time and memory. The new implicit technique was applied to a 3-D supersonic test case on a relatively coarse mesh. For a comprehensive assessment of this technique further test calculations are required.

The paper by *Merkle et al.* [18] was devoted to convergence acceleration of the Navier-Stokes equation through a time-derivative preconditioning of the governing equations. Using physical arguments, a generalized preconditioner was developed, ensuring convergence characteristics which are independent of the Mach number. The uniform convergence was demonstrated for a variety of applications covering a wide range of Mach numbers. In many low speed cases, the preconditioned system showed a much improved convergence rate while having no detrimental effects in regimes where the original method already worked efficiently. So, preconditioning of the governing equations may offer the possibility to develop an efficient unified flow solver for the whole Mach number regime. Further research is required to establish this approach.

At the conference none of the papers devoted to convergence acceleration addressed the key problem of computing realistic Reynolds number flows. These flows require computational meshes with very high aspect ratio or irregular cells leading to very stiff discrete equations. The development of numerical strategies to overcome the stiffness and to ensure fast convergence in these flow situations is one of the grand challenges in algorithmic research.

2.7 Turbulent Flows, LES/DNS

The key problem of accurate numerical simulation of complex flows is the description of transition and turbulence. Currently, in all industrial relevant calculations, the Reynolds averaged Navier-Stokes equations are solved, in which only the statistically stationary flow is calculated and the effects of turbulence are modelled by a so-called turbulence model. However, in many cases the quality of the solution may strongly depend on the turbulence model used in the calculation and at best questionable results may be obtained for more complex flow phenomena such as massive flow separation. The rapid increase of computer resources motivated the research on direct numerical simulation (DNS) or large eddy simulation (LES) of turbulent flows. In the case of DNS, the unsteady Navier-Stokes equations are solved directly. No turbulence model is required since all scales and turbulence motions present are resolved numerically. Due to excessive computer resources required even for simple geometries, this simulation technique is out of question for practical applications. However, it provides a very important methodology for turbulence research. In contrast to DNS, the large eddy simulation of turbulent flows resolves only the large scale structure of the turbulence, while the effects of smaller eddies are described by a statistical subgrid model. As the resolution of the fine scale turbulence motion is not required, far fewer grid points are needed making LES feasible for practical problems at relevant Reynolds numbers in the near future. On the other hand, in order to ensure improved results compared to the solution of the Reynolds averaged Navier-Stokes equations (RANS), besides the establishment of a suitable subgrid model, accurate resolution of the viscous sub-layers in the near wall regions is needed. This substantially increases the number of grid points for LES compared to RANS solvers. Furthermore, since time accurate solutions are calculated in the framework of LES, significant further development of the classical CFD methods is needed. In addition to the validation of a subgrid model, more sophisticated algorithms such as adaptive grids, higher order discretizations, efficient unsteady solvers and parallel computing have to be made available for LES before this technique can be used as a tool for flow simulations.

Numerical aspects of DNS and LES were addressed by papers [5],[20],[21] and [22]. As already mentioned above, the papers [5] and [21] were devoted to the exploitation of parallel computers, whereas paper [20] presented a grid adaption method specially designed for LES. The focus of *Comte et al.* [22] was the investigation of subgrid scale models within a classical explicit finite difference method. The aim of the presentation was to show some examples of what can be achieved with today's supercomputers and standard codes using eddy-viscosity models.

In summary, from the papers delivered at the conference it is very difficult to estimate whether a large eddy simulation for a practical problem, such as a clean wing at a relevant Reynolds number, will become feasible in near future. In order to reach that goal, significant research work on both algorithms and subgrid scale models is needed. A few preliminary approaches for algorithmic improvement were shown at the conference.

2.8 Chemically Reacting Flows

The effective use of CFD for viscous hypersonic reacting flows is one of the present challenges. In the past, substantial effort has been devoted to this research area and key requirements for efficient solution algorithms have been identified [30]. These are sharp capturing of strong shocks, robustness in regions of strong flow expansion, high resolution of viscous regions, efficient treatment of adverse grid and flow situations in the case of complex 3-D geometries, and effective integration of stiff equations introduced by the large chemical source terms.

The two conference papers devoted to reacting flows addressed these algorithmic issues. The paper by *Radespiel et al.* [30] reviewed recent progress made with flux vector splitting methods to ensure high resolution and robustness for hypersonic viscous reacting flow simulations. Two promising approaches recently published in the literature were discussed and compared. Both schemes use scalar dissipation functions and their conceptual differences appear in the resolution of shock waves. Implementation details and recommendations for their effective use for viscous flows were given. Furthermore, the capabilities of the multigrid method based on explicit multistage time-stepping schemes were investigated for reacting flows. A number of modest modifications of the standard multigrid method successfully used for subsonic and transonic flow problems were reported in order to ensure fast convergence for high Mach number flows with strong shocks. The stiffness of the equations introduced by the large chemical source terms is removed by a point implicit treatment. Various computations for different complex flow problems were presented. They impressively demonstrate that with the reported algorithmic improvements converged flow solutions for reacting flows over complex 3-D configurations are now feasible.

The paper devoted by *Coquel et al.* [31] focused on the extension of a hybrid upwind spitting method to non-equilibrium flows. Based on the experience that the classical Van Leer flux vector scheme is not suitable for viscous calculations and the Roe type flux difference solvers are not robust for hypersonic flows, a new upwind approach was presented which basically combines the distinct flux vector and flux difference splitting concepts while retaining their interesting features. The proposed method is a combination of the Van Leer scheme and the Osher scheme with some modifications and extensions. The ability of the new method to resolve viscous hypersonic reacting flows was illustrated by various results including internal and external flow configurations. The time integration is performed by an unfactored implicit scheme, which in the current implementation leads to somewhat slow convergence rate and needs to be improved for further applications.

In conclusion the two papers on reacting flows covered the key issues for developing efficient numerical tools for the simulation of complex flows. Very promising results were presented, illustrating that effective predictions in terms of both accuracy and efficiency for complex configurations are now feasible.

2.9 Unsteady Flows

For steady flows, substantial CFD capability has been achieved over the last two decades and Euler/Navier-Stokes solvers are intensively used in aerodynamic design. In contrast, although some isolated unsteady flow calculations have been carried out for various classes of problems, numerical simulation of unsteady flow fields based on Euler/Navier-Stokes equations is certainly not routine for industrial applications, due to the excessive computational effort involved in these calculations. From the algorithmic point of view, new innovative concepts are required, which substantially cut down the costs of time accurate simulations. This is especially important for viscous flow calculations, where a very fine mesh near the wall is required to resolve the boundary layer. Issues that are central to unsteady CFD are the use of efficient implicit time integration with favorable stability and accuracy characteristics, moving grids, adaptive grids with local grid refinement/coarsening and parallel computing. Moreover, for aeroelastic applications efficient coupling strategies are required.

Time accurate calculations have been addressed by several papers (e.g. [27], [32], [33], [34], [35]). The paper by *Pentaris et al.* [32] focused on the solution of the unsteady incompressible 2-D Navier-Stokes equations using a projection methodology developed for collocated grids. Standard numerical schemes, such as approximate factorization techniques, were employed. The numerical results presented for some test cases were encouraging, however, no remarks on the efficiency of the method were given.

The paper delivered by *Allen* [33] was devoted to grid adaption for unsteady inviscid airfoil flows. The solution adaptive grids are generated by a new transfinite interpolation technique. An interesting approach was presented, in which adaption is performed by adapting the interpolation parameters instead of the physical grid positions. For unsteady calculations, grid adaption is performed gradually by imposing a so-called adaption velocity onto each grid point. The grid interpolation strategy was shown to be well suited for structured moving grids. It is very flexible and requires only little CPU time. Steady and unsteady airfoil computations were presented illustrating the improved results from the adapted meshes. For the calculation, an upwind Euler solver with the dual-time implicit approach was used, which is considerably more efficient than the basic explicit solver. The paper focused on two-dimensional inviscid applications, so that the flexibility and efficiency of the proposed grid adaption strategy are still to be verified for both viscous and three-dimensional flows. A time-varying grid technique was also presented by [26]. Here, the time integration was carried out with a second order implicit scheme.

A more sophisticated moving grid technique was presented by *Jones et al.* [27], with the goal of computing aircraft store trajectories. The technique is based on fully unstructured or hybrid meshes. It was pointed out that the geometric conservation law has to be satisfied within the framework of moving grids in order to guarantee consistent results. So far, only two-dimensional unsteady results have been achieved.

The paper by *Ruis Calavera et al.* [35] addressed parametric studies of a time accurate Euler code for oscillating wings. A rather standard central scheme with artificial dissipation and explicit multistage time stepping scheme was used. Effects of grid density and artificial viscosity on the time accurate solutions were discussed showing the expected behavior. The code has been implemented on a powerful parallel computer, namely the National Wind Tunnel of NAL in Japan. It was demonstrated that parallel computing is a necessary ingredient for effective three-dimensional unsteady flow calculations.

In summary a view central issues for unsteady computations were discussed by the conference papers. However, no major progress in the development of algorithms for efficient three-dimensional time accurate calculations was presented.

3. CONCLUDING REMARKS

In chapter 2 each specific subject of the meeting has already been fully commented, so that only general concluding remarks are given here.

In the evaluator‘s opinion the theme of the symposium "Progress and Challenges in CFD Algorithms and Methods“ was too encompassing and too ambitious for a 3 1/2 day long AGARD conference. Many papers of great interest and high technical standard were delivered. They addressed specific challenges in CFD, proposed new methods or modifications to known methodologies and presented smaller or larger progress. On the other hand, however, quite a large number of papers of lower quality were presented, which either did not focus on current key issues of algorithmic research or mainly reinvented well known results. Probably this situation is very similar to all other large CFD conferences. But measured against the ambitious theme of this symposium, it has to be clearly stated that in many areas the Seville conference did not reflect the actual status of CFD and its recent progress. Considering Jameson‘s excellent survey paper, it is obvious that several important algorithmic developments and recent improvements were not addressed. For example, no paper was devoted to aerodynamic shape optimization and multidisciplinary analysis, topics which are increasingly important for future CFD applications in industry. Furthermore, in some areas such as unstructured grids and adaptive schemes, CFD is much further developed than reported at the conference. Since many leading experts, especially those from the U.S., did not contribute to the conference, it is hard to expect that the high demands of the symposium could be met.

Nevertheless, several important directions of algorithmic research were addressed, which are expected to improve the capability of CFD for complex applications in the industrial environment. These included parallel computing, advanced discretization techniques, fast iterative solvers and powerful acceleration techniques, adaptive schemes and flexible strategies for discretizing the computational domain. Interesting and new aspects of these techniques were discussed, substantiating their extended potentials and improved abilities. In most cases, however, the superiority of the more sophisticated methods to the well established standard schemes was only demonstrated for simplified test problems, for which the classical methods also perform quite well. Very often results were shown for 2-D inviscid and laminar viscous flows. Three-dimensional calculations were restricted in most cases to inviscid flows or simplified geometries. Only a few more realistic calculations were presented. To make a step forward, it is very important to apply the advanced methodologies to those problems, for which the standard methods show substantial deficiencies in terms of accuracy and efficiency or do not work at all. One of the grand challenges in CFD is the effective simulation of viscous flows at realistic, full scale Reynolds numbers for complex configurations. This problem, although ideal for testing advanced discretiztion and time integration schemes, was hardly tackled at the conference even for simplified geometries. Furthermore, in order to raise the confidence level of CFD methods, careful grid refinement studies, sensitivity investigations, estimation and control of the numerical error as well as detailed code validation are required for a wide class of relevant applications. In many papers, these issues were only partly or not all considered.

In conclusion, considerable research work is still needed to establish CFD as an effective tool in the aerodynamic design process. The most important, but probably also the most limiting factor, is turbulence modelling, a subject which was outside the scope of this symposium. With respect to algorithms, further development and improvement remain essential but have to be directed towards the real challenges in CFD, which include:

- accurate viscous flow simulation at relevant Reynolds numbers
- effective treatment of complex configurations, such as a complete aircraft
- efficient simulation of more complex flows with multiple space and time scales, such as unsteady flows or reacting flows
- large eddy simulation for practical applications
- aerodynamic shape optimization
- multidisciplinary analysis and design

The Seville symposium was a step in the right direction. For some topics, it showed some good promise but there is still considerable work to be done to meet the challenges of industrial CFD. The symposium provided a valuable forum for exchange of information about recent developments and achievements.

4. LITERATURE

1. Jameson, A., "Present Status, Challenges and Future Developments in CFD".
2. Rubbert, P., "CFD Research in the Changing U.S. Aeronautical Industry".
3. Knight, D., "Parallel Computing in CFD".
4. Eisfeld, B., Ritzdorf, H., Bleecke, H.M., Kroll, N., "Portable Parallelization of a 3-D Euler/Navier Stokes Solver for Complex Flows".
5. Pinelli, A., Vacca, A., "A Parallel Spectral Multidomain Solver for Incompressible Navier-Stokes Equations".
6. Wissink, A., Lyrintzis, A., Strawn, R., "On Improving Parallelism in the Transonic Unsteady Rotor Navier-Stokes Code TURNS".
7. Dias D'Almeida, F., Castro, F. A., Palma, J.M.L. M., Vasconcelos, P., "Development of a Parallel Implicit Algorithm for CFD Calculations".
8. Ramakrishnan, S.V., Szema, K.Y., Chen, C.L., Shankar, V.V., Chakravarthy, S.R., "Experiments with Unstructured Grid Computations".
9. Delanaye, M., Geuzaine, Ph., Essers, J.A., "A Secon Order Accurate Finite Volume Scheme Solving Euler and Navier-Stokes Equations on Adaptive Unstructured Grids with an Implicit Convergence Acceleration Procedure".
10. Villedieu, Ph., Estivalezes, J. L., "A New Positivity Preserving Second Order Accurate Kinetic Scheme for the Multidimensional Euler Equations on Unstructured Meshes".
11. Onate, E., Fischer, J., "Meshless Techniques for Computer Analysis of High Speed Flows".
12. Pirumov, U.G., Kryukov, I.A., Ivanov, I.E., "Numerical Simulation of Internal and External Gas Dynamic Flows on Structured and Unstructured Adaptive Grids".
13. Briggs, R. D., Shahpar, S., "An Investigation of the Effects of the Artivicial Dissipation Terms in a Modern TVD Scheme on the Solution of a Viscous Flow Problem".
14. Vinckier, A., Jacobsen, J., Wagner, S., "Multidimensional Upwinding with Flux Filters for the Euler and Navier-Stokes Equations".
15. Paillère, P., Carette, J.C., Issman, E., Van der Weide, E., Deconinck, H., Degrez, G. "Implicit Multidimensional Upwind Residual Distribution Schemes on Adaptive Meshes".
16. Van Ransbeeck, P., Hirsch, Ch., "Multidimensional Upwind Dissipation for 2-D/3-D Euler/Navier-Stokes Applications".

17. Aslan, A.R., Gülcat, U., Misirrhoglu, A., "A PCG/E-B-E Iteration for High Order and Fast Solution of 3-D Navier-Stokes Equations".

18. Merkle, C.L., Venkateswaran, S., "Convergence Acceleration of Navier-Stokes Computations Through Time Derivative Preconditioning".

19. Pulliam, T.H., Rogers, S. Barth, T., "Practical Aspects of Krylov Subspace Based-Iterative Methods in CFD".

20. Van der Vegt, J.J.W., Van der Ven, H., "Hexahedron Based Grid Adaption Algorithm for Future Large Eddy Simulation".

21. Streng, M., Kuerten, H., Broeze, J., Geurts, B., "Parallel Algorithms for DNS of Compressible Flow".

22. Comte, P., "A Straightforward 3-D Multi-Block Unsteady Navier-Stokes Solver for Direct and Large Eddy Simulations of Transitional and Turbulent Compressible Flows".

23. Orszag, S.A., Quian, Y.H., Succi, S., "Applications of Lattice Boltzmann Methods to Fluid Mechanics".

24. Becker, K., Rill, S., "Structured Adaptive Sub-Block Refinement for 3-D Flows".

25. Sibilla, S., Vitaletti, M., "Multiblock Structured Grid Algorithms for Euler Solvers in a Parallel Computing Framework".

26. Cambier, L., Darracq, D., Gazaix, M., Guillen, Ph., Jouet, Ch., Le Touellec, L., "Améliorations Récentes du Code de Calcul d'Ecoulements Compressibles FLU3M".

27. Jones, D.J., Fortin, F., Syms, G.F., Hawken, D., "The Computation of Aircraft Store Trajectories using Hybrid (Structured/Unstructured) Grids".

28. Galle, M., "Solution of the Euler and Navier-Stokes Equations on Hybrid Grids".

29. Brenner, P., "Simulation du Mouvement Relatif de Corps Soumis à un Ecoulement Instaionnaire par une Méthode de Chevauchement de Maillages".

30. Radespiel, R., Longo, J.M.A., Brück S., Schwamborn, D., "Efficient Numerical Simulation of Complex 3-D Flows with Large Contrast".

31. Coquel, F., Joly, V., Marmignon, C., "Méthods de Décentrement Hybrides pour la Simulation d'Ecoulements en Déséquilibre Thermique et Chimique".

32. Pentaris, A., Tsangaris, S., "A Projection Methodology for the Simulation of Unstaedy Incompressible Viscous Flows Using the Approximate Factorization Technique".

33. Allen, C.B., "An Implicit Upwind Scheme with Grid Adaption for Unsteady Euler Aerofoil Flows".

34. Friedrichs, O., Gerhold, Th., Meister, A., Sonar, Th., "Adaptive Computation of Unsteady Flow Fields with DLR-τ-Code".

35. Ruiz Calavera, R.P., Hirose, N., "Parametric Studies of a Time Accurate Finite Volume Euler Code in the N.W.T. Parallel Computer".

36. Badcock, K.J., Richards, B.E., "Parallel Implicit Upwind Methods for the Aerodynamics of Aerospace Vehicles".

The Present Status, Challenges, and Future Developments in Computational Fluid Dynamics

Antony Jameson
Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, New Jersey 08544 USA

1. SUMMARY

This paper presents a perspective on computational fluid dynamics as a tool for aircraft design. It addresses the requirements for effective industrial use, and trade-offs between modelling accuracy and computational costs. Issues in algorithm design are discussed in detail, together with a unified approach to the design of shock capturing algorithms. Finally, the paper discusses the use of techniques drawn from control theory to determine optimal aerodynamic shapes. In the future multidisciplinary analysis and optimization should be combined to provide an integrated numerical design environment.

2. INTRODUCTION

Computational methods first began to have a significant impact on aerodynamic analysis and design in the period of 1965-75. This decade saw the introduction of panel methods which could solve the linear flow models for arbitrarily complex geometry in both subsonic and supersonic flow [58, 147, 179]. It also saw the appearance of the first satisfactory methods for treating the nonlinear equations of transonic flow [123, 122, 63, 64, 43, 54], and the development of the hodograph method for the design of shock free supercritical airfoils [15].

Computational Fluid Dynamics (CFD) has now matured to the point at which it is widely accepted as a key tool for aerodynamic design. Algorithms have been the subject of intensive development for the past two decades. The principles underlying the design and implementation of robust schemes which can accurately resolve shock waves and contact discontinuities in compressible flows are now quite well established. It is also quite well understood how to design high order schemes for viscous flow, including compact schemes and spectral methods. Adaptive refinement of the mesh interval (h) and the order of approximations (p) has been successfully exploited both separately and in combination in the h-p method [126]. A continuing obstacle to the treatment of configurations with complex geometry has been the problem of mesh generation. Several general techniques have been developed, including algebraic transformations and methods based on the solution of elliptic and hyperbolic equations. In the last few years methods using unstructured meshes have also begun to gain more general acceptance. The Dassault-INRIA group led the way in developing a finite element method for transonic potential flow. They obtained a solution for a complete Falcon 50 as early as 1982 [25]. Euler methods for unstructured meshes have been the subject of intensive development by several groups since 1985 [110, 82, 81, 163, 14], and Navier-Stokes methods on unstructured meshes have also been demonstrated [117, 118, 11].

Despite the advances that have been made, CFD is still not being exploited as effectively as one would like in the design process. This is partly due to the long set-up and high costs, both human and computational of complex flow simulations. The essential requirements for industrial use are:

1. assured accuracy
2. acceptable computational and human costs
3. fast turn around.

Improvements are still needed in all three areas. In particular, the fidelity of modelling of high Reynolds number viscous flows continues to be limited by computational costs. Consequently accurate and cost-effective simulation of viscous flow at Reynolds numbers associated with full scale flight, such as the prediction of high lift devices, remains a challenge. Several routes are available toward the reduction of computational costs, including the reduction of mesh requirements by the use of higher order schemes, improved convergence to a steady state by sophisticated acceleration methods, fast inversion methods for implicit schemes, and the exploitation of massively parallel computers.

Another factor limiting the effective use of CFD is the lack of good interfaces to computer aided design (CAD) systems. The geometry models provided by existing CAD systems often fail to meet the requirements of continuity and smoothness needed for flow simulation, with the consequence that they must be modified before they can be used to provide the input for mesh generation. This bottleneck, which impedes the automation of the mesh generation process, needs to be eliminated, and the CFD software should be fully integrated in a numerical design environment. In addition to more accurate and cost-effective flow prediction methods, better optimizations methods are also needed, so that not only can designs be rapidly evaluated, but directions of improvement can be identified. Possession of techniques which result in a faster design cycle gives a crucial advantage in a competitive environment.

A critical issue, examined in the next section, is the choice of mathematical models. What level of complexity is needed to provide sufficient accuracy for aerodynamic design, and what is the impact on cost and turn-around time? Section 3 addresses the design of numerical algorithms for flow simulation. Section 4 presents the results of some numerical calculations which require moderate computer resources and could be completed with the fast turn-around required by industrial users. Section 5 discusses automatic design procedures which can be used to produce optimum aerodynamic designs. Finally, Section 7. offers an outlook for the future.

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

3. THE COMPLEXITY OF FLUID FLOW AND MATHEMATICAL MODELLING

3.1 The Hierarchy of Mathematical Models

Many critical phenomena of fluid flow, such as shock waves and turbulence, are essentially non-linear. They also exhibit extreme disparities of scales. While the actual thickness of a shock wave is of the order of a mean free path of the gas particles, on a macroscopic scale its thickness is essentially zero. In turbulent flow energy is transferred from large scale motions to progressively smaller eddies until the scale becomes so small that the motion is dissipated by viscosity. The ratio of the length scale of the global flow to that of the smallest persisting eddies is of the order $\mathrm{Re}^{\frac{3}{4}}$, where Re is the Reynolds number, typically in the range of 30 million for an aircraft. In order to resolve such scales in all three space directions a computational grid with the order of $\mathrm{Re}^{\frac{9}{4}}$ cells would be required. This is beyond the range of any current or foreseeable computer. Consequently mathematical models with varying degrees of simplification have to be introduced in order to make computational simulation of flow feasible, and to produce viable and cost-effective methods.

Figure 1 (supplied by Pradeep Raj) indicates a hierarchy of models at different levels of simplification which have proved useful in practice. Efficient flight is generally achieved by the use of smooth and streamlined shapes which avoid flow separation and minimize viscous effects, with the consequence that useful predictions can be made using inviscid models. Inviscid calculations with boundary layer corrections can provide quite accurate predictions of lift and drag when the flow remains attached, but iteration between the inviscid outer solution and the inner boundary layer solution becomes increasingly difficult with the onset of separation. Procedures for solving the full viscous equations are likely to be needed for the simulation of arbitrary complex separated flows, which may occur at high angles of attack or with bluff bodies. In order to treat flows at high Reynolds numbers, one is generally forced to estimate turbulent effects by Reynolds averaging of the fluctuating components. This requires the introduction of a turbulence model. As the available computing power increases one may also aspire to large eddy simulation (LES) in which the larger scale eddies are directly calculated, while the influence of turbulence at scales smaller than the mesh interval is represented by a subgrid scale model.

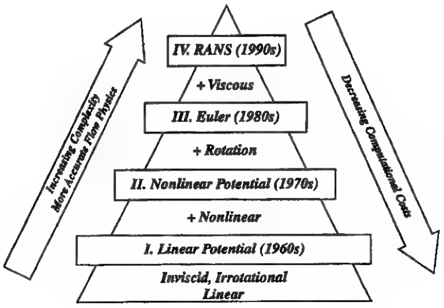

Increasing Complexity
More Accurate Flow Physics
IV. RANS (1990s)
+ Viscous
III. Euler (1980s)
+ Rotation
II. Nonlinear Potential (1970s)
+ Nonlinear
I. Linear Potential (1960s)
Inviscid, Irrotational
Linear
Decreasing Computational Costs

Figure 1: Hierarchy of Fluid Flow Models

3.2 Computational Costs

Computational costs vary drastically with the choice of mathematical model. Panel methods can be effectively used to solve the linear potential flow equation with higher-end personal computers (with an Intel 80486 microprocessor, for example). Studies of the dependency of the result on mesh refinement, performed by this author and others, have demonstrated that inviscid transonic potential flow or Euler solutions for an airfoil can be accurately calculated on a mesh with 160 cells around the section, and 32 cells normal to the section. Using multigrid techniques 10 to 25 cycles are enough to obtain a converged result. Consequently airfoil calculations can be performed in seconds on a Cray YMP, and can also be performed on 486-class personal computers. Correspondingly accurate three-dimensional inviscid calculations can be performed for a wing on a mesh, say with $192\times32\times48$=294,912 cells, in about 5 minutes on a single processor Cray YMP, or less than a minute with eight processors, or in 1 or 2 hours on a workstation such as a Hewlett Packard 735 or an IBM 560 model.

Viscous simulations at high Reynolds numbers require vastly greater resources. Careful two-dimensional studies of mesh requirements have been carried out at Princeton by Martinelli [114]. He found that on the order of 32 mesh intervals were needed to resolve a turbulent boundary layer, in addition to 32 intervals between the boundary layer and the far field, leading to a total of 64 intervals. In order to prevent degradations in accuracy and convergence due to excessively large aspect ratios (in excess of 1,000) in the surface mesh cells, the chordwise resolution must also be increased to 512 intervals. Reasonably accurate solutions can be obtained in a 512×64 mesh in 100 multigrid cycles. Translated to three dimensions, this would imply the need for meshes with 5–10 million cells (for example, $512\times64\times256$ = 8,388,608 cells as shown in Figure 2). When simulations are performed on less fine meshes with, say, 500,000 to 1 million cells, it is very hard to avoid mesh dependency in the solutions as well as sensitivity to the turbulence model.

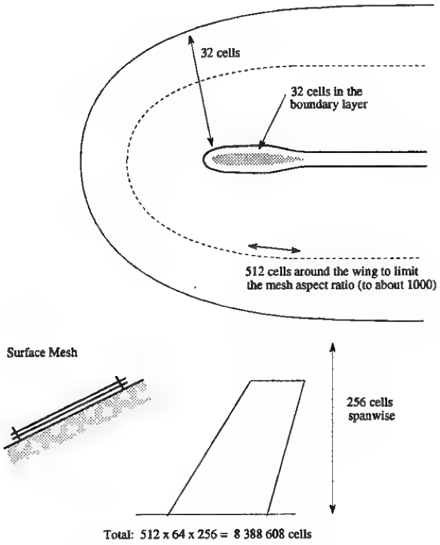

32 cells
32 cells in the boundary layer
512 cells around the wing to limit the mesh aspect ratio (to about 1000)
Surface Mesh
256 cells spanwise
Total: 512 x 64 x 256 = 8 388 608 cells

Figure 2: Mesh Requirements for a Viscous Simulation

A typical algorithm requires of the order of 5,000 floating point operations per mesh point in one multigrid iteration. With 10 million mesh points, the operation count is of the order of 0.5×10^{11} per cycle. Given a computer capable of sustaining 10^{11} operations per second (100 gigaflops), 200 cycles could then be performed in 100 seconds. Simulations of unsteady viscous flows (flutter, buffet) would be likely to require 1,000–10,000 time steps. A further progression to large eddy simulation of complex configurations would require even greater resources. The following estimate is due to W.H. Jou [90]. Suppose that a conservative estimate of the size of eddies in a boundary layer that ought to be resolved is $1/5$ of the boundary layer thickness. Assuming that 10 points are needed to resolve a single eddy, the mesh interval should then be $1/50$ of the boundary layer thickness. Moreover, since the eddies are three-dimensional, the same mesh interval should be used in all three directions. Now, if the boundary layer thickness is of the order of 0.01 of the chord length, 5,000 intervals will be needed in the chordwise direction, and for a wing with an aspect ratio of 10, 50,000 intervals will be needed in the spanwise direction. Thus, of the order of $50 \times 5,000 \times 50,000$ or 12.5 billion mesh points would be needed in the boundary layer. If the time dependent behavior of the eddies is to be fully resolved using time steps on the order of the time for a wave to pass through a mesh interval, and one allows for a total time equal to the time required for waves to travel three times the length of the chord, of the order of 15,000 time steps would be needed. Performance beyond the teraflop (10^{12} operations per second) will be needed to attempt calculations of this nature, which also have an information content far beyond what is needed for enginering analysis and design. The designer does not need to know the details of the eddies in the boundary layer. The primary purpose of such calculations is to improve the prediction of averaged quantities such as skin friction, and the prediction of global behavior such as the onset of separation. The main current use of Navier-Stokes and large eddy simulations is to gain an improved insight into the physics of turbulent flow, which may in turn lead to the development of more comprehensive and reliable turbulence models.

3.3 Turbulence Modelling

It is doubtful whether a universally valid turbulence model, capable of describing all complex flows, could be devised [52]. Algebraic models [30, 9] have proved fairly satisfactory for the calculation of attached and slightly separated wing flows. These models rely on the boundary layer concept, usually incorporating separate formulas for the inner and outer layers, and they require an estimate of a length scale which depends on the thickness of the boundary layer. The estimation of this quantity by a search for a maximum of the vorticity times a distance to the wall, as in the Baldwin-Lomax model, can lead to ambiguities in internal flows, and also in complex vortical flows over slender bodies and highly swept or delta wings [40, 115]. The Johnson-King model [88], which allows for non-equilibrium effects through the introduction of an ordinary differential equation for the maximum shear stress, has improved the prediction of flows with shock induced separation [148, 91].

Closure models depending on the solution of transport equations are widely accepted for industrial applications. These models eliminate the need to estimate a length scale by detecting the edge of the boundary layer. Eddy viscosity models typically use two equations for the turbulent kinetic energy k and the dissipation rate ϵ, or a pair of equivalent quantities [89, 178, 160, 1, 121, 35]. Models of this type generally tend to present difficulties in the region very close to the wall. They also tend to be badly conditioned for numerical solution. The $k-l$ model [154] is designed to alleviate this problem by taking advantage of the linear behaviour of the length scale l near the wall. In an alternative approach to the design of models which are more amenable to numerical solution, new models requiring the solution of one transport equation have recently been introduced [10, 159]. The performance of the algebraic models remains competitive for wing flows, but the one- and two-equation models show promise for broader classes of flows. In order to achieve greater universality, research is also being pursued on more complex Reynolds stress transport models, which require the solution of a larger number of transport equations.

Another direction of research is the attempt to devise more rational models via renormalization group (RNG) theory [182, 155]. Both algebraic and two-equation $k-\epsilon$ models devised by this approach have shown promising results [116].

The selection of sufficiently accurate mathematical models and a judgment of their cost-effectiveness ultimately rests with industry. Aircraft and spacecraft designs normally pass through the three phases of conceptual design, preliminary design, and detailed design. Correspondingly, the appropriate CFD models will vary in complexity. In the conceptual and preliminary design phases, the emphasis will be on relatively simple models which can give results with very rapid turn-around and low computer costs, in order to evaluate alternative configurations and perform quick parametric studies. The detailed design stage requires the most complete simulation that can be achieved with acceptable cost. In the past, the low level of confidence that could be placed on numerical predictions has forced the extensive use of wind tunnel testing at an early stage of the design. This practice was very expensive. The limited number of models that could be fabricated also limited the range of design variations that could be evaluated. It can be anticipated that in the future, the role of wind tunnel testing in the design process will be more one of verification. Experimental research to improve our understanding of the physics of complex flows will continue, however, to play a vital role.

4. CFD ALGORITHMS

4.1 Difficulties of Flow Simulation

The computational simulation of fluid flow presents a number of severe challenges for algorithm design. At the level of inviscid modeling, the inherent nonlinearity of the fluid flow equations leads to the formation of singularities such as shock waves and contact discontinuities. Moreover, the geometric configurations of interest are extremely complex, and generally contain sharp edges which lead to the shedding of vortex sheets. Extreme gradients near stagnation points or wing tips may also lead to numerical errors that can have global influence. Numerically generated entropy may be convected from the leading edge, for example, causing the formation of a numerically induced boundary layer which can lead to separation. The need to treat exterior domains of infinite extent is also a source of difficulty. Boundary conditions imposed at artificial outer boundaries may cause reflected waves which significantly interfere with the flow. When viscous effects are also included in the simulation, the extreme difference of the scales in the viscous boundary layer and the outer flow, which is essentially inviscid, is another source of difficulty, forcing the use of meshes with extreme variations in the mesh intervals. For these reasons, CFD has been a driving force for the development of numerical algorithms.

4.2 Structured and Unstructured Meshes

The algorithm designer faces a number of critical decisions. The first choice that must be made is the nature of the mesh used to divide the flow field into discrete subdomains. The discretization procedure must allow for the treatment of complex configurations. The principal alternatives are Cartesian meshes, body-fitted curvilinear meshes, and unstructured tetrahedral meshes. Each of these approaches has advantages which have led to their use. The Cartesian mesh minimizes the complexity of the algorithm at interior points and facilitates the use of high order discretization procedures, at the expense of greater complexity, and possibly a loss of accuracy, in the treatment of boundary conditions at curved surfaces. This difficulty may be alleviated by using mesh refinement procedures near the surface. With their aid, schemes which use Cartesian meshes have recently been developed to treat very complex configurations [120, 149, 22, 94].

Body-fitted meshes have been widely used and are particularly well suited to the treatment of viscous flow because they readily allow the mesh to be compressed near the body surface. With this approach, the problem of mesh generation itself has proved to be a major pacing item. The most commonly used procedures are algebraic transformations [7, 44, 46, 156], methods based on the solution of elliptic equations, pioneered by Thompson [170, 171, 157, 158], and methods based on the solution of hyperbolic equations marching out from the body [161]. In order to treat very complex configurations it generally proves expedient to use a multiblock [177, 150] procedure, with separately generated meshes in each block, which may then be patched at block faces, or allowed to overlap, as in the Chimera scheme [19, 20]. While a number of interactive software systems for grid generation have been developed, such as EAGLE, GRIDGEN, and ICEM, the generation of a satisfactory grid for a very complex configuration may require months of effort.

The alternative is to use an unstructured mesh in which the domain is subdivided into tetrahedra. This in turn requires the development of solution algorithms capable of yielding the required accuracy on unstructured meshes. This approach has been gaining acceptance, as it is becoming apparent that it can lead to a speed-up and reduction in the cost of mesh generation that more than offsets the increased complexity and cost of the flow simulations. Two competing procedures for generating triangulations which have both proved successful are Delaunay triangulation [41, 11], based on concepts introduced at the beginning of the century by Voronoi [175], and the moving front method [111].

4.3 Finite Difference, Finite Volume, and Finite Element Schemes

Associated with choice of mesh type is the formulation of the discretization procedure for the equations of fluid flow, which can be expressed as differential conservation laws. In the Cartesian tensor notation, let x_i be the coordinates, p, ρ, T, and E the pressure, density, temperature, and total energy, and u_i the velocity components. Using the convention that summation over j=1, 2, 3 is implied by a repeated subscript j, each conservation equation has the form

$$\frac{\partial w}{\partial t}+\frac{\partial f_j}{\partial x_j}=0. \qquad (1)$$

For the mass equation

$$w=\rho,\quad f_j=\rho u_j.$$

For the i momentum equation

$$w_i=\rho u_i,\quad f_{ij}=\rho u_i u_j+p\delta_{ij}-\sigma_{ij},$$

where σ_{ij} is the viscous stress tensor. For the energy equation

$$w=\rho E,\quad f_j=(\rho E+p)\,u_j-\sigma_{jk}u_k-\kappa\frac{\partial T}{\partial x_j},$$

where κ is the coefficient of heat conduction. The pressure is related to the density and energy by the equation of state

$$p=(\gamma-1)\,\rho\left(E-\frac{1}{2}u_i u_i\right) \qquad (2)$$

in which γ is the ratio of specific heats. In the Navier-Stokes equations the viscous stresses are assumed to be linearly proportional to the rate of strain, or

$$\sigma_{ij}=\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)+\lambda\delta_{ij}\left(\frac{\partial u_k}{\partial x_k}\right), \qquad (3)$$

where μ and λ are the coefficients of viscosity and bulk viscosity, and usually $\lambda=-2\mu/3$.

The finite difference method, which requires the use of a Cartesian or a structured curvilinear mesh, directly approximates the differential operators appearing in these equations. In the finite volume method [112], the discretization is accomplished by dividing the domain of the flow into a large number of small subdomains, and applying the conservation laws in the integral form

$$\frac{\partial}{\partial t}\int_\Omega w dV+\int_{\partial\Omega}\mathbf{f}\cdot d\mathbf{S}=0.$$

Here $\mathbf{f}$ is the flux appearing in equation (1) and $d\mathbf{S}$ is the directed surface element of the boundary $\partial\Omega$ of the domain Ω. The use of the integral form has the advantage that no assumption of the differentiability of the solutions is implied, with the result that it remains a valid statement for a subdomain containing a shock wave. In general the subdomains could be arbitrary, but it is convenient to use either hexahedral cells in a body conforming curvilinear mesh or tetrahedrons in an unstructured mesh.

Alternative discretization schemes may be obtained by storing flow variables at either the cell centers or the vertices. These variations are illustrated in Figure 3 for the two-dimensional case. With a cell-centered scheme the discrete conservation law takes the form

$$\frac{d}{dt}wV+\sum_{\text{faces}}\mathbf{f}\cdot\mathbf{S}=0, \qquad (4)$$

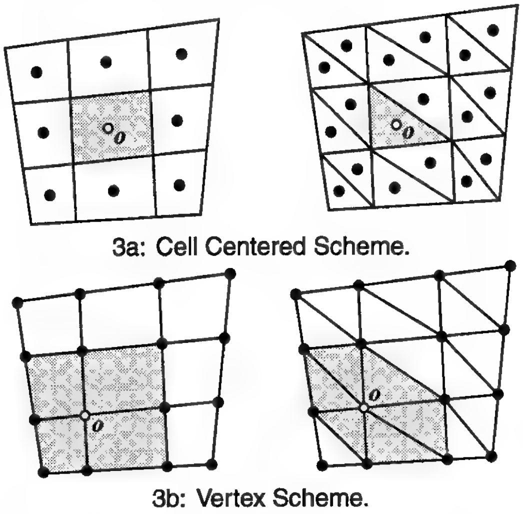


Figure 3: Structured and Unstructured Discretizations.

where V is the cell volume, and $\mathbf{f}$ is now a numerical estimate of the flux vector through each face. $\mathbf{f}$ may be evaluated from values of the flow variables in the cells separated by each face, using upwind biasing to allow for the directions of wave propagation. With hexahedral cells, equation (4) is very similar to a finite difference scheme in curvilinear coordinates. Under a transformation to curvilinear coordinates ξ_j, equation (1) becomes

$$\frac{\partial}{\partial t}(Jw)+\frac{\partial}{\partial\xi_i}\left(J\frac{\partial\xi_i}{\partial x_j}f_j\right)=0, \qquad (5)$$

where J is the Jacobian determinant of the transformation matrix $\left[\frac{\partial x_i}{\partial\xi_j}\right]$. The transformed flux $J\frac{\partial\xi_i}{\partial x_j}f_j$ corresponds to the dot product of the flux $\mathbf{f}$ with a vector face area $J\frac{\partial\xi_i}{\partial x_j}$, while J represents the transformation of the cell volume. The finite volume form (4) has the advantages that it is valid for both structured and unstructured meshes, and that it assures that a uniform flow exactly satisfies the equations, because $\sum_{\text{faces}}\mathbf{S}=0$ for a closed control volume. Finite difference schemes do not necessarily satisfy this constraint because of the discretization errors in evaluating $\frac{\partial \xi_i}{\partial x_j}$ and the inversion of the transformation matrix. A cell-vertex finite volume scheme can be derived by taking the union of the cells surrounding a given vertex as the control volume for that vertex [55, 71, 139]. In equation (4), V is now the sum of the volumes of the surrounding cells, while the flux balance is evaluated over the outer faces of the polyhedral control volume. In the absence of upwind biasing the flux vector is evaluated by averaging over the corners of each face. This has the advantage of remaining accurate on an irregular or unstructured mesh. An alternative route to the discrete equations is provided by the finite element method. Whereas the finite difference and finite volume methods approximate the differential and integral operators, the finite element method proceeds by inserting an approximate solution into the exact equations. On multiplying by a test function ϕ and integrating by parts over space, one obtains the weak form

$$\frac{\partial}{\partial t}\iiint_{\Omega} \phi w d\Omega = \iiint_{\Omega} \mathbf{f}\cdot\nabla\phi d\Omega - \iint_{\partial\Omega} \phi \mathbf{f}\cdot d\mathbf{S} \qquad (6)$$

which is also valid in the presence of discontinuities in the flow. In the Galerkin method the approximate solution is expanded in terms of the same family of functions as those from which the test functions are drawn. By choosing test functions with local support, separate equations are obtained for each node. For example, if a tetrahedral mesh is used, and ϕ is piecewise linear, with a nonzero value only at a single node, the equations at each node have a stencil which contains only the nearest neighbors. In this case the finite element approximation corresponds closely to a finite volume scheme. If a piecewise linear approximation to the flux $\mathbf{f}$ is used in the evaluation of the integrals on the right hand side of equation (6), these integrals reduce to formulas which are identical to the flux balance of the finite volume scheme.

Thus the finite difference and finite volume methods lead to essentially similar schemes on structured meshes, while the finite volume method is essentially equivalent to a finite element method with linear elements when a tetrahedral mesh is used. Provided that the flow equations are expressed in the conservation law form (1), all three methods lead to an exact cancellation of the fluxes through interior cell boundaries, so that the conservative property of the equations is preserved. The important role of this property in ensuring correct shock jump conditions was pointed out by Lax and Wendroff [97].

4.4 Non-oscillatory Shock Capturing Schemes

4.4.1 Local Extremum Diminishing (LED) Schemes

The discretization procedures which have been described in the last section lead to nondissipative approximations to the Euler equations. Dissipative terms may be needed for two reasons. The first is the possibility of undamped oscillatory modes. The second reason is the need for the clean capture of shock waves and contact discontinuities without undesirable oscillations. An extreme overshoot could result in a negative value of an inherently positive quantity such as the pressure or density. The next sections summarize a unified approach to the construction of nonoscillatory schemes via the introduction of controlled diffusive and antidiffusive terms. This is the line adhered to in the author's own work.

The development of non-oscillatory schemes has been a prime focus of algorithm research for compressible flow. Consider a general semi-discrete scheme of the form

$$\frac{d}{dt}v_j = \sum_{k \neq j} c_{jk}\left(v_k - v_j\right). \qquad (7)$$

A maximum cannot increase and a minimum cannot decrease if the coefficients c_{jk} are non-negative, since at a maximum $v_k - v_j \leq 0$, and at a minimum $v_k - v_j \geq 0$. Thus the condition

$$c_{jk} \geq 0, \quad k \neq j \qquad (8)$$

is sufficient to ensure stability in the maximum norm. Moreover, if the scheme has a compact stencil, so that c_{jk}=0 when j and k are not nearest neighbors, a local maximum cannot increase and local minimum cannot decrease. This local extremum diminishing (LED) property prevents the birth and growth of oscillations. The one-dimensional conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0$$

provides a useful model for analysis. In this case waves are propagated with a speed $a(u) = \frac{\partial f}{\partial u}$, and the solution is constant along the characteristics $\frac{dx}{dt} = a(u)$. Thus the LED property is satisfied. In fact the total variation

$$TV(u) = \int_{-\infty}^{\infty} \left|\frac{\partial u}{\partial x}\right| dx$$

of a solution of this equation does not increase, provided that any discontinuity appearing in the solution satisfies an entropy condition [96]. Harten proposed that difference schemes ought to be designed so that the discrete total variation cannot increase [56]. If the end values are fixed, the total variation can be expressed as

$$TV(u) = 2\left(\sum \text{maxima} - \sum \text{minima}\right).$$

Thus a LED scheme is also total variation diminishing (TVD). Positivity conditions of the type expressed in equations (7) and (8) lead to diagonally dominant schemes, and are the key to the elimination of improper oscillations. The positivity conditions may be realized by the introduction of diffusive terms or by the use of upwind biasing in the discrete scheme. Unfortunately, they may also lead to severe restrictions on accuracy unless the coefficients have a complex nonlinear dependence on the solution.

4.4.2 Artificial Diffusion and Upwinding

Following the pioneering work of Godunov [51], a variety of dissipative and upwind schemes designed to have good shock capturing properties have been developed during the past two decades [162, 23, 98, 100, 146, 130, 56, 129, 166, 5, 68, 183, 62, 180, 13, 12, 11]. If the one-dimensional scalar conservation law

$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x} f(v) = 0 \qquad (9)$$

is represented by a three point scheme

$$\frac{dv_j}{dt} = c^{+}_{j+\frac{1}{2}}\left(v_{j+1} - v_j\right) + c^{-}_{j-\frac{1}{2}}\left(v_{j-1} - v_j\right),$$

the scheme is LED if

$$c^{+}_{j+\frac{1}{2}} \geq 0, \quad c^{-}_{j-\frac{1}{2}} \geq 0. \qquad (10)$$

A conservative semidiscrete approximation to the one-dimensional conservation law can be derived by subdividing the line into cells. Then the evolution of the value v_j in the jth cell is given by

$$\Delta x \frac{dv_j}{dt} + h_{j+\frac{1}{2}} - h_{j-\frac{1}{2}} = 0, \qquad (11)$$

where $h_{j+\frac{1}{2}}$ is an estimate of the flux between cells j and $j+1$. The simplest estimate is the arithmetic average $(f_{j+1}+f_j)/2$, but this leads to a scheme that does not satisfy the positivity conditions. To correct this, one may add a dissipative term and set

$$h_{j+\frac{1}{2}}=\frac{1}{2}\left(f_{j+1}+f_j\right)-\alpha_{j+\frac{1}{2}}\left(v_{j+1}-v_j\right). \qquad (12)$$

In order to estimate the required value of the coefficient $\alpha_{j+\frac{1}{2}}$, let $a_{j+\frac{1}{2}}$ be a numerical estimate of the wave speed $\frac{\partial f}{\partial u}$,

$$a_{j+\frac{1}{2}}=\begin{cases} \frac{f_{j+1}-f_j}{v_{j+1}-v_j} & \text{if} \quad v_{j+1}\neq v_j \\ \left.\frac{\partial f}{\partial v}\right|_{v=v_j} & \text{if} \quad v_{j+1}=v_j \end{cases}. \qquad (13)$$

Then

$$\begin{aligned} h_{j+\frac{1}{2}}-h_{j-\frac{1}{2}} &= +\left(\frac{1}{2}a_{j+\frac{1}{2}}-\alpha_{j+\frac{1}{2}}\right)\Delta v_{j+\frac{1}{2}} \\ &\quad +\left(\frac{1}{2}a_{j-\frac{1}{2}}+\alpha_{j-\frac{1}{2}}\right)\Delta v_{j-\frac{1}{2}}, \end{aligned}$$

where

$$\Delta v_{j+\frac{1}{2}}=v_{j+1}-v_j,$$

and the LED condition (10) is satisfied if

$$\alpha_{j+\frac{1}{2}}\geq\frac{1}{2}\left|a_{j+\frac{1}{2}}\right|. \qquad (14)$$

If one takes

$$\alpha_{j+\frac{1}{2}}=\frac{1}{2}\left|a_{j+\frac{1}{2}}\right|,$$

one obtains the first order upwind scheme

$$h_{j+\frac{1}{2}}=\begin{cases} f_j & \text{if } a_{j+\frac{1}{2}}>0 \\ f_{j+1} & \text{if } a_{j+\frac{1}{2}}<0 \end{cases}.$$

This is the least diffusive first order scheme which satisfies the LED condition. In this sense upwinding is a natural approach to the construction of non-oscillatory schemes. It may be noted that the successful treatment of transonic potential flow also involved the use of upwind biasing. This was first introduced by Murman and Cole to treat the transonic small disturbance equation [123].

Another important requirement of discrete schemes is that they should exclude nonphysical solutions which do not satisfy appropriate entropy conditions [95], which require the convergence of characteristics towards admissible discontinuities. This places more stringent bounds on the minimum level of numerical viscosity [113, 169, 128, 131]. In the case that the numerical flux function is strictly convex, Aiso has recently proved [2] that it is sufficient that

$$\alpha_{j+\frac{1}{2}}>\max\left\{\frac{1}{2}\left|a_{j+\frac{1}{2}}\right|,\epsilon\,\text{sign}(v_{j+1}-v_j)\right\}$$

for $\epsilon>0$. Thus the numerical viscosity should be rounded out and not allowed to reach zero at a point where the wave speed $a(u)=\frac{\partial f}{\partial u}$ approaches zero. This justifies, for example, Harten's entropy fix [56].

Higher order schemes can be constructed by introducing higher order diffusive terms. Unfortunately these have larger stencils and coefficients of varying sign which are not compatible with the conditions (8) for a LED scheme, and it is known that schemes which satisfy these conditions are at best first order accurate in the neighborhood of an extremum. It proves useful in the following development to introduce the concept of essentially local extremum diminishing (ELED) schemes. These are defined to be schemes which satisfy the condition that in the limit as the mesh width $\Delta x\to 0$, local maxima are non-increasing, and local minima are non-decreasing.

4.4.3 High Resolution Switched Schemes: Jameson-Schmidt-Turkel (JST) Scheme

Higher order non-oscillatory schemes can be derived by introducing anti-diffusive terms in a controlled manner. An early attempt to produce a high resolution scheme by this approach is the Jameson-Schmidt-Turkel (JST) scheme [85]. Suppose that anti-diffusive terms are introduced by subtracting neighboring differences to produce a third order diffusive flux

$$d_{j+\frac{1}{2}}=\alpha_{j+\frac{1}{2}}\left\{\Delta v_{j+\frac{1}{2}}-\frac{1}{2}\left(\Delta v_{j+\frac{3}{2}}+\Delta v_{j-\frac{1}{2}}\right)\right\}, \qquad (15)$$

which is an approximation to $\frac{1}{2}\alpha\Delta x^3\frac{\partial^3}{\partial x^3}$. The positivity condition (8) is violated by this scheme. It proves that it generates substantial oscillations in the vicinity of shock waves, which can be eliminated by switching locally to the first order scheme. The JST scheme therefore introduces blended diffusion of the form

$$\begin{aligned} d_{j+\frac{1}{2}} &= +\epsilon^{(2)}_{j+\frac{1}{2}}\Delta v_{j+\frac{1}{2}} \\ &\quad -\epsilon^{(4)}_{j+\frac{1}{2}}\left(\Delta v_{j+\frac{3}{2}}-2\Delta v_{j+\frac{1}{2}}+\Delta v_{j-\frac{1}{2}}\right), \end{aligned} \qquad (16)$$

The idea is to use variable coefficients $\epsilon^{(2)}_{j+\frac{1}{2}}$ and $\epsilon^{(4)}_{j+\frac{1}{2}}$ which produce a low level of diffusion in regions where the solution is smooth, but prevent oscillations near discontinuities. If $\epsilon^{(2)}_{j+\frac{1}{2}}$ is constructed so that it is of order Δx^2 where the solution is smooth, while $\epsilon^{(4)}_{j+\frac{1}{2}}$ is of order unity, both terms in $d_{j+\frac{1}{2}}$ will be of order Δx^3.

The JST scheme has proved very effective in practice in numerous calculations of complex steady flows, and conditions under which it could be a total variation diminishing (TVD) scheme have been examined by Swanson and Turkel [165]. An alternative statement of sufficient conditions on the coefficients $\epsilon^{(2)}_{j+\frac{1}{2}}$ and $\epsilon^{(4)}_{j+\frac{1}{2}}$ for the JST scheme to be LED is as follows:

Theorem 1 (Positivity of the JST scheme) *Suppose that whenever either v_{j+1} or v_j is an extremum the coefficients of the JST scheme satisfy*

$$\epsilon^{(2)}_{j+\frac{1}{2}}\geq\frac{1}{2}\left|\alpha_{j+\frac{1}{2}}\right|,\ \epsilon^{(4)}_{j+\frac{1}{2}}=0. \qquad (17)$$

Then the JST scheme is local extremum diminishing (LED).

Proof: *We need only consider the rate of change of v at extremal points. Suppose that v_j is an extremum. Then*

$$\epsilon^{(4)}_{j+\frac{1}{2}}=\epsilon^{(4)}_{j-\frac{1}{2}}=0,$$

and the semi-discrete scheme (11) reduces to

$$\begin{aligned} \Delta x\frac{dv_j}{dt} &= \left(\epsilon^{(2)}_{j+\frac{1}{2}}-\frac{1}{2}a_{j+\frac{1}{2}}\right)\Delta v_{j+\frac{1}{2}} \\ &\quad -\left(\epsilon^{(2)}_{j-\frac{1}{2}}+\frac{1}{2}a_{j-\frac{1}{2}}\right)\Delta v_{j-\frac{1}{2}}, \end{aligned}$$

and each coefficient has the required sign. □

In order to construct $\epsilon^{(2)}_{j-\frac{1}{2}}$ and $\epsilon^{(4)}_{j-\frac{1}{2}}$ with the desired properties define

$$R(u,v)=\begin{cases} \left|\frac{u-v}{|u|+|v|}\right|^q & \text{if } u \not=0 \text{ or } v \not=0 \\ 0 & \text{if } u=v=0, \end{cases} \qquad (18)$$

where q is a positive integer. Then $R(u,v)=1$ if u and v have opposite signs. Otherwise $R(u,v) < 1$. Now set

$$Q_j=R(\Delta v_{j+\frac{1}{2}}, \Delta v_{j-\frac{1}{2}}), \quad Q_{j+\frac{1}{2}}=\max(Q_j, Q_{j+1}).$$

and

$$\epsilon^{(2)}_{j+\frac{1}{2}}=\alpha_{j+\frac{1}{2}} Q_{j+\frac{1}{2}}, \quad \epsilon^{(4)}_{j+\frac{1}{2}}=\frac{1}{2}\alpha_{j+\frac{1}{2}}(1-Q_{j+\frac{1}{2}}). \qquad (19)$$

4.4.4 Symmetric Limited Positive (SLIP) Scheme

An alternative route to high resolution without oscillation is to introduce flux limiters to guarantee the satisfaction of the positivity condition (8). The use of limiters dates back to the work of Boris and Book [23]. A particularly simple way to introduce limiters, proposed by the author in 1984 [68], is to use flux limited dissipation. In this scheme the third order diffusion defined by equation (15) is modified by the insertion of limiters which produce an equivalent three point scheme with positive coefficients. The original scheme [68] can be improved in the following manner so that less restrictive flux limiters are required. Let $L(u,v)$ be a limited average of u and v with the following properties:

P1. $L(u,v)=L(v,u)$

P2. $L(\alpha u,\alpha v)=\alpha L(u,v)$

P3. $L(u,u)=u$

P4. $L(u,v)=0$ if u and v have opposite signs: otherwise $L(u,v)$ has the same sign as u and v.

Properties (P1–P3) are natural properties of an average. Property (P4) is needed for the construction of a LED or TVD scheme.

It is convenient to introduce the notation

$$\phi(r)=L(1,r)=L(r,1),$$

where according to (P4) $\phi(r) \geq 0$. It follows from (P2) on setting $\alpha=\frac{1}{u}$ or $\frac{1}{v}$ that

$$L(u,v)=\phi\left(\frac{v}{u}\right)u=\phi\left(\frac{u}{v}\right)v.$$

Also it follows on setting v=1 and u=r that

$$\phi(r)=r\phi\left(\frac{1}{r}\right).$$

Thus, if there exists $r < 0$ for which $\phi(r) > 0$, then $\phi\left(\frac{1}{r}\right) < 0$. The only way to ensure that $\phi(r) \geq 0$ is to require $\phi(r)=0$ for all $r < 0$, corresponding to property (P4).

Now one defines the diffusive flux for a scalar conservation law as

$$d_{j+\frac{1}{2}}=\alpha_{j+\frac{1}{2}}\left\{\Delta v_{j+\frac{1}{2}} - L\left(\Delta v_{j+\frac{3}{2}}, \Delta v_{j-\frac{1}{2}}\right)\right\}. \qquad (20)$$

Set

$$r^+=\frac{\Delta v_{j+\frac{3}{2}}}{\Delta v_{j-\frac{1}{2}}}, \quad r^-=\frac{\Delta v_{j-\frac{3}{2}}}{\Delta v_{j+\frac{1}{2}}}.$$

and

$$\begin{aligned} L(\Delta v_{j+\frac{3}{2}}, \Delta v_{j-\frac{1}{2}}) &= \phi(r^+)\,\Delta v_{j-\frac{1}{2}} \\ L(\Delta v_{j-\frac{3}{2}}, \Delta v_{j+\frac{1}{2}}) &= \phi(r^-)\,\Delta v_{j+\frac{1}{2}}. \end{aligned}$$

Then,

$$\begin{aligned} \Delta x \tfrac{dv_j}{dt} &= \left\{\alpha_{j+\frac{1}{2}} - \frac{1}{2}a_{j+\frac{1}{2}} + \alpha_{j-\frac{1}{2}}\phi(r^-)\right\}\Delta v_{j+\frac{1}{2}} \\ - & \left\{\alpha_{j-\frac{1}{2}} + \frac{1}{2}a_{j-\frac{1}{2}} + \alpha_{j+\frac{1}{2}}\phi(r^+)\right\}\Delta v_{j-\frac{1}{2}}. \end{aligned} \qquad (21)$$

Thus the scheme satisfies the LED condition if $\alpha_{j+\frac{1}{2}} \geq \frac{1}{2}\left|a_{j+\frac{1}{2}}\right|$ for all j, and $\phi(r) \geq 0$, which is assured by property (P4) on L. At the same time it follows from property (P3) that the first order diffusive flux is canceled when Δv is smoothly varying and of constant sign. Schemes constructed by this formulation will be referred to as symmetric limited positive (SLIP) schemes. This result may be summarized as

Theorem 2 (Positivity of the SLIP scheme)
Suppose that the discrete conservation law (11) contains a limited diffusive flux as defined by equation (20). Then the positivity condition (14), together with the properties (P1–P4) for limited averages, are sufficient to ensure satisfaction of the LED principle that a local maximum cannot increase and a local minimum cannot decrease. □

A variety of limiters may be defined which meet the requirements of properties (P1–P4). Define

$$S(u,v)=\frac{1}{2}\{\text{sign}(u)+\text{sign}(v)\}$$

which vanishes is u and v have opposite signs.

Then two limiters which are appropriate are the following well-known schemes:

1. Minmod:

$$L(u,v)=S(u,v)\min(|u|,|v|)$$

2. Van Leer:

$$L(u,v)=S(u,v)\,\frac{2|u||v|}{|u|+|v|}.$$

In order to produce a family of limiters which contains these as special cases it is convenient to set

$$L(u,v)=\frac{1}{2}D(u,v)\,(u+v),$$

where $D(u,v)$ is a factor which should deflate the arithmetic average, and become zero if u and v have opposite signs. Take

$$D(u,v)=1-R(u,v)=1-\left|\frac{u-v}{|u|+|v|}\right|^q, \qquad (22)$$

where $R(u,v)$ is the same function that was introduced in the JST scheme, and q is a positive integer. Then $D(u,v)=0$ if u and v have opposite signs. Also if q=1, $L(u,v)$ reduces to minmod, while if q=2, $L(u,v)$ is equivalent to Van Leer's limiter. By increasing q one can generate a sequence of limited averages which approach a limit defined by the arithmetic mean truncated to zero when u and v have opposite signs.

When the terms are regrouped, it can be seen that with this limiter the SLIP scheme is exactly equivalent to the JST scheme, with the switch is defined as

$$
\begin{aligned}
Q_{j+\frac{1}{2}} &= R\left(\Delta v_{j+\frac{3}{2}}, \Delta v_{j+\frac{1}{2}}\right) \\
\epsilon^{(2)}_{j+\frac{1}{2}} &= \alpha_{j+\frac{1}{2}} Q_{j+\frac{1}{2}} \\
\epsilon^{(4)}_{j+\frac{1}{2}} &= \alpha_{j+\frac{1}{2}} \left(1 - Q_{j+\frac{1}{2}}\right).
\end{aligned}
$$

This formulation thus unifies the JST and SLIP schemes.

4.4.5 *Essentially Local Extremum Diminishing (ELED) Scheme with Soft Limiter*

The limiters defined by the formula (22) have the disadvantage that they are active at a smooth extrema, reducing the local accuracy of the scheme to first order. In order to prevent this, the SLIP scheme can be relaxed to give an essentially local extremum diminishing (ELED) scheme which is second order accurate at smooth extrema by the introduction of a threshold in the limited average. Therefore redefine $D(u,v)$ as

$$
D(u,v) = 1 - \left| \frac{u - v}{\max(|u| + |v|, \epsilon \Delta x^r)} \right|^q, \tag{23}
$$

where $r = \frac{3}{2}$, $q \geq 2$. This reduces to the previous definition if $|u| + |v| > \epsilon \Delta x^r$.

In any region where the solution is smooth, $\Delta v_{j+\frac{3}{2}} - \Delta v_{j-\frac{1}{2}}$ is of order Δx^2. In fact if there is a smooth extremum in the neighborhood of v_j or v_{j+1}, a Taylor series expansion indicates that $\Delta v_{j+\frac{3}{2}}$, $\Delta v_{j+\frac{1}{2}}$ and $\Delta v_{j-\frac{1}{2}}$ are each individually of order Δx^2, since $\frac{dv}{dx}=0$ at the extremum. It may be verified that second order accuracy is preserved at a smooth extremum if $q \geq 2$. On the other hand the limiter acts in the usual way if $\left|\Delta v_{j+\frac{3}{2}}\right|$ or $\left|\Delta v_{j-\frac{3}{2}}\right| > \epsilon \Delta x^r$, and it may also be verified that in the limit $\Delta x \to 0$ local maxima are non increasing and local minima are non decreasing [79]. Thus the scheme is essentially local extremum diminishing (ELED).

The effect of the "soft limiter" is not only to improve the accuracy: the introduction of a threshold below which extrema of small amplitude are accepted also usually results in a faster rate of convergence to a steady state, and decreases the likelyhood of limit cycles in which the limiter interacts unfavorably with the corrections produced by the updating scheme. In a scheme recently proposed by Venkatakrishnan a threshold is introduced precisely for this purpose [174].

4.4.6 *Upstream Limited Positive (USLIP) Schemes*

By adding the anti-diffusive correction purely from the upstream side one may derive a family of upstream limited positive (USLIP) schemes. Corresponding to the original SLIP scheme defined by equation (20), a USLIP scheme is obtained by setting

$$
d_{j+\frac{1}{2}} = \alpha_{j+\frac{1}{2}} \left\{ \Delta v_{j+\frac{1}{2}} - L\left(\Delta v_{j+\frac{1}{2}}, \Delta v_{j-\frac{1}{2}}\right) \right\}
$$

if $a_{j+\frac{1}{2}} > 0$, or

$$
d_{j+\frac{1}{2}} = \alpha_{j+\frac{1}{2}} \left\{ \Delta v_{j+\frac{1}{2}} - L\left(\Delta v_{j+\frac{1}{2}}, \Delta v_{j+\frac{3}{2}}\right) \right\}
$$

if $a_{j+\frac{1}{2}} < 0$. If $\alpha_{j+\frac{1}{2}} = \frac{1}{2}\left|a_{j+\frac{1}{2}}\right|$ one recovers a standard high resolution upwind scheme in semi-discrete form. Consider the case that $a_{j+\frac{1}{2}} > 0$ and $a_{j-\frac{1}{2}} > 0$. If one sets

$$
r^+ = \frac{\Delta v_{j+\frac{1}{2}}}{\Delta v_{j-\frac{1}{2}}}, \quad r^- = \frac{\Delta v_{j-\frac{3}{2}}}{\Delta v_{j-\frac{1}{2}}},
$$

the scheme reduces to

$$
\Delta x \frac{dv_j}{dt} = -\frac{1}{2} \left\{ \phi(r^+)\, a_{j+\frac{1}{2}} + \left(2 - \phi(r^-)\right) a_{j-\frac{1}{2}} \right\} \Delta v_{j-\frac{1}{2}}.
$$

To assure the correct sign to satisfy the LED criterion the flux limiter must now satisfy the additional constraint that $\phi(r) \leq 2$.

The USLIP formulation is essentially equivalent to standard upwind schemes [130, 166]. Both the SLIP and USLIP constructions can be implemented on unstructured meshes [75, 79]. The anti-diffusive terms are then calculated by taking the scalar product of the vectors defining an edge with the gradient in the adjacent upstream and downstream cells.

4.4.7 *Systems of Conservation Laws: Flux Splitting and Flux-Difference Splitting*

Steger and Warming [162] first showed how to generalize the concept of upwinding to the system of conservation laws

$$
\frac{\partial w}{\partial t} + \frac{\partial}{\partial x} f(w) = 0 \tag{24}
$$

by the concept of flux splitting. Suppose that the flux is split as $f = f^+ + f^-$ where $\frac{\partial f^+}{\partial w}$ and $\frac{\partial f^-}{\partial w}$ have positive and negative eigenvalues. Then the first order upwind scheme is produced by taking the numerical flux to be

$$
h_{j+\frac{1}{2}} = f^+_j + f^-_{j+1}.
$$

This can be expressed in viscosity form as

$$
\begin{aligned}
h_{j+\frac{1}{2}} &= \frac{1}{2}\left(f^+_{j+1} + f^+_j\right) - \frac{1}{2}\left(f^+_{j+1} - f^+_j\right) \\
&\quad + \frac{1}{2}\left(f^-_{j+1} + f^-_j\right) + \frac{1}{2}\left(f^-_{j+1} - f^-_j\right) \\
&= \frac{1}{2}\left(f_{j+1} + f_j\right) - d_{j+\frac{1}{2}},
\end{aligned}
$$

where the diffusive flux is

$$
d_{j+\frac{1}{2}} = \frac{1}{2} \Delta (f^+ - f^-)_{j+\frac{1}{2}}. \tag{25}
$$

Roe derived the alternative formulation of flux difference splitting [146] by distributing the corrections due to the flux difference in each interval upwind and downwind to obtain

$$
\Delta x \frac{dw_j}{dt} + (f_{j+1} - f_j)^- + (f_j - f_{j-1})^+ = 0,
$$

where now the flux difference $f_{j+1} - f_j$ is split. The corresponding diffusive flux is

$$
d_{j+\frac{1}{2}} = \frac{1}{2}\left(\Delta f^+_{j+\frac{1}{2}} - \Delta f^-_{j+\frac{1}{2}}\right).
$$

Following Roe's derivation, let $A_{j+\frac{1}{2}}$ be a mean value Jacobian matrix exactly satisfying the condition

$$
f_{j+1} - f_j = A_{j+\frac{1}{2}} (w_{j+1} - w_j). \tag{26}
$$

$A_{j+\frac{1}{2}}$ may be calculated by substituting the weighted averages

$$u=\frac{\sqrt{\rho_{j+1}}u_{j+1}+\sqrt{\rho_j}u_j}{\sqrt{\rho_{j+1}}+\sqrt{\rho_j}},\ H=\frac{\sqrt{\rho_{j+1}}H_{j+1}+\sqrt{\rho_j}H_j}{\sqrt{\rho_{j+1}}+\sqrt{\rho_j}} \qquad (27)$$

into the standard formulas for the Jacobian matrix $A=\frac{\partial f}{\partial w}$. A splitting according to characteristic fields is now obtained by decomposing $A_{j+\frac{1}{2}}$ as

$$A_{j+\frac{1}{2}}=T\Lambda T^{-1}, \qquad (28)$$

where the columns of T are the eigenvectors of $A_{j+\frac{1}{2}}$, and Λ is a diagonal matrix of the eigenvalues. Now the corresponding diffusive flux is

$$\frac{1}{2}\left|A_{j+\frac{1}{2}}\right|(w_{j+1}-w_j),$$

where

$$\left|A_{j+\frac{1}{2}}\right|=T\,|\Lambda|\,T^{-1}$$

and $|\Lambda|$ is the diagonal matrix containing the absolute values of the eigenvalues.

4.4.8 *Alternative Splittings*

Characteristic splitting has the advantages that it introduces the minimum amount of diffusion to exclude the growth of local extrema of the characteristic variables, and that with the Roe linearization it allows a discrete shock structure with a single interior point. To reduce the computational complexity one may replace $|A|$ by αI where if α is at least equal to the spectral radius $\max|\lambda(A)|$, then the positivity conditions will still be satisfied. Then the first order scheme simply has the scalar diffusive flux

$$d_{j+\frac{1}{2}}=\frac{1}{2}\alpha_{j+\frac{1}{2}}\Delta w_{j+\frac{1}{2}}. \qquad (29)$$

The JST scheme with scalar diffusive flux captures shock waves with about 3 interior points, and it has been widely used for transonic flow calculations because it is both robust and computationally inexpensive.

An intermediate class of schemes can be formulated by defining the first order diffusive flux as a combination of differences of the state and flux vectors

$$d_{j+\frac{1}{2}}=\frac{1}{2}\alpha^*_{j+\frac{1}{2}}c\left(w_{j+1}-w_j\right)+\frac{1}{2}\beta_{j+\frac{1}{2}}\left(f_{j+1}-f_j\right), \qquad (30)$$

where the factor c is included in the first term to make $\alpha^*_{j+\frac{1}{2}}$ and $\beta_{j+\frac{1}{2}}$ dimensionless. Schemes of this class are fully upwind in supersonic flow if one takes $\alpha^*_{j+\frac{1}{2}}=0$ and $\beta_{j+\frac{1}{2}}=\text{sign}(M)$ when the absolute value of the Mach number M exceeds 1. The flux vector f can be decomposed as

$$f=uw+f_p, \qquad (31)$$

where

$$f_p=\begin{pmatrix} 0 \\ p \\ up \end{pmatrix}. \qquad (32)$$

Then

$$f_{j+1}-f_j=\bar{u}\left(w_{j+1}-w_j\right)+\bar{w}\left(u_{j+1}-u_j\right)+f_{p_{j+1}}-f_{p_j}, \qquad (33)$$

where $\bar{u}$ and $\bar{w}$ are the arithmetic averages

$$\bar{u}=\frac{1}{2}\left(u_{j+1}+u_j\right),\quad \bar{w}=\frac{1}{2}\left(w_{j+1}+w_j\right).$$

Thus these schemes are closely related to schemes which introduce separate splittings of the convective and pressure terms, such as the wave-particle scheme [141, 8], the advection upwind splitting method (AUSM) [106, 176], and the convective upwind and split pressure (CUSP) schemes [76].

In order to examine the shock capturing properties of these various schemes, consider the general case of a first order diffusive flux of the form

$$d_{j+\frac{1}{2}}=\frac{1}{2}\alpha_{j+\frac{1}{2}}B_{j+\frac{1}{2}}\left(w_{j+1}-w_j\right), \qquad (34)$$

where the matrix $B_{j+\frac{1}{2}}$ determines the properties of the scheme and the scaling factor $\alpha_{j+\frac{1}{2}}$ is included for convenience. All the previous schemes can be obtained by representing $B_{j+\frac{1}{2}}$ as a polynomial in the matrix $A_{j+\frac{1}{2}}$ defined by equation (26). Schemes of this class were considered by Van Leer [99]. According to the Cayley-Hamilton theorem, a matrix satisfies its own characteristic equation. Therefore the third and higher powers of A can be eliminated, and there is no loss of generality in limiting $B_{j+\frac{1}{2}}$ to a polynomial of degree 2,

$$B_{j+\frac{1}{2}}=\alpha_0 I+\alpha_1 A_{j+\frac{1}{2}}+\alpha_2 A^2_{j+\frac{1}{2}}. \qquad (35)$$

The characteristic upwind scheme for which $B_{j+\frac{1}{2}}=\left|A_{j+\frac{1}{2}}\right|$ is obtained by substituting $A_{j+\frac{1}{2}}=T\Lambda T^{-1}$, $A^2_{j+\frac{1}{2}}=T\Lambda^2 T^{-1}$. Then α_0, α_1, and α_2 are determined from the three equations

$$\alpha_0+\alpha_1\lambda_k+\alpha_2\lambda_k^2=|\lambda_k|,\quad k=1,2,3.$$

The same representation remains valid for three dimensional flow because $A_{j+\frac{1}{2}}$ still has only three distinct eigenvalues u, $u+c$, $u-c$.

4.4.9 *Analysis of Stationary Discrete Shocks*

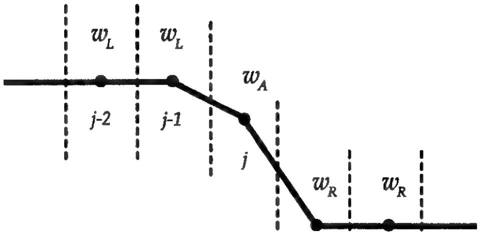


Figure 4: Shock structure for single interior point.

The ideal model of a discrete shock is illustrated in figure (4). Suppose that w_L and w_R are left and right states which satisfy the jump conditions for a stationary shock, and that the corresponding fluxes are $f_L=f(w_L)$ and $f_R=f(w_R)$. Since the shock is stationary $f_L=f_R$. The ideal discrete shock has constant states w_L to the left and w_R to the right, and a single point with an intermediate value w_A. The intermediate value is needed to allow the discrete solution to correspond to a true solution in which the shock wave does not coincide with an interface between two mesh cells.

Schemes corresponding to one, two or three terms in equation (35) are examined in [80]. The analysis of these three cases shows that a discrete shock structure with a single interior point is supported by artificial diffusion that satisfies the two conditions that

1. it produces an upwind flux if the flow is determined to be supersonic through the interface
2. it satisfies a generalized eigenvalue problem for the exit from the shock of the form

$$(A_{AR} - \alpha_{AR} B_{AR})(w_R - w_A)=0, \qquad (36)$$

where A_{AR} is the linearized Jacobian matrix and B_{AR} is the matrix defining the diffusion for the interface AR. This follows from the equilibrium condition $h_{RA}=h_{RR}$ for the cell $j+1$ in figure 4. These two conditions are satisfied by both the characteristic scheme and also the CUSP scheme, provided that the coefficients of convective diffusion and pressure differences are correctly balanced. Scalar diffusion does not satisfy the first condition. In the case of the CUSP scheme (30) equation (36) reduces to

$$\left(A_{RA} + \frac{\alpha^* c}{1+\beta}\right)(w_R - w_A)=0$$

Thus $w_R - w_A$ is an eigenvector of the Roe matrix A_{RA}, and $-\frac{\alpha^* c}{1+\beta}$ is the corresponding eigenvalue. Since the eigenvalues are u, $u+c$, and $u-c$, the only choice which leads to positive diffusion when $u > 0$ is $u-c$, yielding the relationship

$$\alpha^* c=(1+\beta)\,(c-u)\,, 0 < u < c$$

Thus there is a one parameter family of schemes which support the ideal shock structure. The term $\beta(f_R - f_A)$ contributes to the diffusion of the convective terms. Allowing for the split (31), the total effective coefficient of convective diffusion is $\alpha c=\alpha^* c + \beta \bar{u}$. A CUSP scheme with low numerical diffusion is then obtained by taking $\alpha=|M|$, leading to the coefficients illustrated in figure 5.

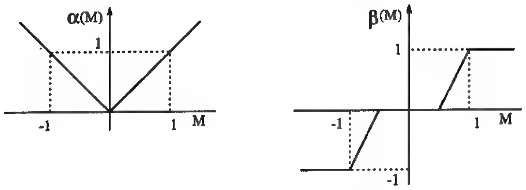


Figure 5: Diffusion Coefficients.

4.4.10 *CUSP and Characteristic Schemes Admitting Constant Total Enthalpy in Steady Flow*

In steady flow the stagnation enthalpy H is constant, corresponding to the fact that the energy and mass conservation equations are consistent when the constant factor H is removed from the energy equation. Discrete and semi-discrete schemes do not necessarily satisfy this property. In the case of a semi-discrete scheme expressed in viscosity form, equations (11) and (12), a solution with constant H is admitted if the viscosity for the energy equation reduces to the viscosity for the continuity equation with ρ replaced by ρH. When the standard characteristic decomposition (28) is used, the viscous fluxes for ρ and ρH which result from composition of the fluxes for the characteristic variables do not have this property, and H is not constant in the discrete solution. In practice there is an excursion of H in the discrete shock structure which represents a local heat source. In very high speed flows the corresponding error in the temperature may lead to a wrong prediction of associated effects such as chemical reactions.

The source of the error in the stagnation enthalpy is the discrepancy between the convective terms

$$u\begin{pmatrix} \rho \\ \rho u \\ \rho H \end{pmatrix},$$

in the flux vector, which contain ρH, and the state vector which contains ρE. This may be remedied by introducing a modified state vector

$$w_h=\begin{pmatrix} \rho \\ \rho u \\ \rho H \end{pmatrix}.$$

Then one introduces the linearization

$$f_R - f_L=A_h(w_{h_R} - w_{h_L})\,.$$

Here A_h may be calculated in the same way as the standard Roe linearization. Introduce the weighted averages defined by equation (27). Then

$$A_h=\begin{pmatrix} 0 & 1 & 0 \\ -\frac{\gamma+1}{\gamma}\frac{u^2}{2} & \frac{\gamma+1}{\gamma}u & \frac{\gamma-1}{\gamma} \\ -uH & H & u \end{pmatrix}.$$

The eigenvalues of A_h are u, λ^+ and λ^- where

$$\lambda^{\pm}=\frac{\gamma+1}{2\gamma}u \pm \sqrt{(\frac{\gamma+1}{2\gamma}u)^2 + \frac{c^2-u^2}{\gamma}}. \qquad (37)$$

Now both CUSP and characteristic schemes which preserve constant stagnation enthalpy in steady flow can be constructed from the modified Jacobian matrix A_h [80]. These schemes also produce a discrete shock structure with one interior point in steady flow. Then one arrives at four variations with this property, which can conveniently be distinguished as the E- and H-CUSP schemes, and the E- and H-characteristic schemes.

4.5 Multidimensional Schemes

The simplest approach to the treatment of multi-dimensional problems on structured meshes is to apply the one-dimensional construction separately in each mesh direction. On triangulated meshes in two or three dimensions the SLIP and USLIP constructions may also be implemented along the mesh edges [79]. A substantial body of current research is directed toward the implementation of truly multi-dimensional upwind schemes in which the upwind biasing is determined by properties of the flow rather than the mesh. A thorough review is given by Pailliere and Deconinck in reference [132].

Residual distribution schemes are an attractive approach for triangulated meshes. In these the residual defined by the space derivatives is evaluated for each cell, and then distributed to the vertices with weights which depend on the direction of convection. For a scalar conservation law the weights can be chosen to maintain positivity with minimum cross diffusion in the direction normal to the flow. For the Euler equations the residual can be linearized by assuming that the parameter vector with components $\sqrt{\rho}$,$\sqrt{\rho}u_i$, and $\sqrt{\rho}H$ varies linearly over the cell. Then

$$\frac{\partial f_j(w)}{\partial x_j}=A_j\frac{\partial w}{\partial x_j}$$

where the Jacobian matrices $A_j=\frac{\partial f_j}{\partial w}$ are evaluated with Roe averaging of the values of w at the vertices. Waves in the direction $\mathbf{n}$ can then be expressed in terms of the eigenvectors of $n_j A_j$, and a positive distribution scheme is used for waves in preferred directions. The best choice of these directions is the subject of ongoing research, but preliminary results indicate the possibility of achieving high resolution of shocks and contact discontinuities which are not aligned with mesh lines [132].

Hirsch and Van Ransbeeck adopt an alternative approach in which they directly construct directional diffusive terms on structured meshes, with anti-diffusion controlled by limiters based on comparisons of slopes in different directions [60]. They also show promising results in calculations of nozzles with multiply reflected oblique shocks.

4.5.1 *High Order Godunov Schemes, and Kinetic Flux Splitting*

A substantial body of current research is directed toward the implementation of truly multi-dimensional upwind schemes [59, 135, 101]. Reference [132] provides a thorough review of recent developments in this field. Some of the most impressive simulations of time dependent flows with strong shock waves have been achieved with higher order Godunov schemes [180]. In these schemes the average value in each cell is updated by applying the integral conservation law using interface fluxes predicted from the exact or approximate solution of a Riemann problem between adjacent cells. A higher order estimate of the solution is then reconstructed from the cell averages, and slope limiters are applied to the reconstruction. An example is the class of essentially non-oscillatory (ENO) schemes, which can attain a very high order of accuracy at the cost of a substantial increase in computational complexity [32, 153, 151, 152]. Methods based on reconstruction can also be implemented on unstructured meshes [13, 12]. Recently there has been an increasing interest in kinetic flux splitting schemes, which use solutions of the Boltzmann equation or the BGK equation to predict the interface fluxes [42, 36, 45, 136, 181].

4.6 Discretization of the Viscous Terms

The discretization of the viscous terms of the Navier Stokes equations requires an approximation to the velocity derivatives $\frac{\partial u_i}{\partial x_j}$ in order to calculate the tensor σ_{ij}, defined by equation (3). Then the viscous terms may be included in the flux balance (4). In order to evaluate the derivatives one may apply the Gauss formula to a control volume V with the boundary S

$$\int_V \frac{\partial u_i}{\partial x_j} dv = \int_S u_i n_j dS,$$

where n_j is the outward normal. For a tetrahedral or hexahedral cell this gives

$$\overline{\frac{\partial u_i}{\partial x_j}} = \frac{1}{\text{vol}} \sum_{\text{faces}} \bar{u}_i \, n_j \, S \qquad (38)$$

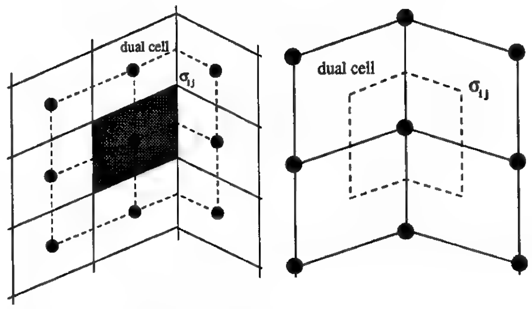
where $\bar{u}_i$ is an estimate of the average of u_i over the face. If u varies linearly over a tetrahedral cell this is exact. Alternatively, assuming a local transformation to computational coordinates ξ_j, one may apply the chain rule

$$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} = \left[\frac{\partial \mathbf{u}}{\partial \xi}\right] \left[\frac{\partial \xi}{\partial \mathbf{x}}\right] = \frac{\partial \mathbf{u}}{\partial \xi} \left[\frac{\partial \mathbf{x}}{\partial \xi}\right]^{-1} \qquad (39)$$

Here the transformation derivatives $\frac{\partial x_i}{\partial \xi_j}$ can be evaluated by the same finite difference formulas as the velocity derivatives $\frac{\partial u_i}{\partial \xi_j}$ In this case $\frac{\partial u}{\partial \xi}$ is exact if u is a linearly varying function.

For a cell-centered discretization (figure 6a) $\frac{\partial u_i}{\partial \xi_j}$ is needed at each face. The simplest procedure is to evaluate $\frac{\partial u_i}{\partial \xi_j}$ in each cell, and to average $\frac{\partial u_i}{\partial \xi_j}$ between the two cells on either side of a face [87]. The resulting discretization does not have a compact stencil, and supports undamped oscillatory modes. In a one-dimensional calculation, for example, $\frac{\partial^2 u}{\partial x^2}$ would be discretized as $\frac{u_{i+2}-2u_i+u_{i-2}}{4\Delta x^2}$. In order to produce a compact stencil $\frac{\partial u_i}{\partial x_j}$ may be estimated from a control volume centered on each face, using formulas (38) or (39) [144]. This is computationally expensive because the number of faces is much larger than the number of cells. In a hexahedral mesh with a large number of vertices the number of faces approaches three times the number of cells.

This motivates the introduction of dual meshes for the evaluation of the velocity derivatives and the flux balance as sketched in figure 6. The figure shows both cell-centered and cell-vertex schemes. The dual mesh connects cell centers of the primary mesh. If there is a kink in the primary mesh, the dual cells should be formed by assembling contiguous fractions of the neighboring primary cells. On smooth meshes comparable results are obtained by either of these formulations [114, 115, 107]. If the mesh has a kink the cell-vertex scheme has the advantage that the derivatives $\frac{\partial u_i}{\partial x_j}$ are calculated in the interior of a regular cell, with no loss of accuracy.



6a: Cell-centered scheme. σ_{ij} evaluated at vertices of the primary mesh

6b: Cell-vertex scheme. σ_{ij} evaluated at cell centers of the primary mesh

Figure 6: Viscous discretizations for cell-centered and cell-vertex algorithms.

A desirable property is that a linearly varying velocity distribution, as in a Couette flow, should produce a constant stress and hence an exact stress balance. This property is not necessarily satisfied in general by finite difference or finite volume schemes on curvilinear meshes. The characterization k-exact has been proposed for schemes that are exact for polynomials of degree k. The cell-vertex finite volume scheme is linearly exact if the derivatives are evaluated by equation (39), since then $\frac{\partial u_i}{\partial x_j}$ is exactly evaluated as a constant, leading to constant viscous stresses σ_{ij}, and an exact viscous stress balance. This remains true when there is a kink in the mesh, because the summation of constant stresses over the faces of the kinked control volume sketched in figure 6 still yields a perfect balance. The use of equation (39) to evaluate $\frac{\partial u_i}{\partial x_j}$, however, requires the additional calculation or storage of the nine metric quantities $\frac{\partial u_i}{\partial x_j}$ in each cell, whereas equation (38) can be evaluated from the same face areas that are used for the flux balance.

In the case of an unstructured mesh, the weak form (6) leads to a natural discretization with linear elements, in which the piecewise linear approximation yields a constant stress in each cell. This method yields a representation which is globally correct when averaged over the cells, a result that can be proved by energy estimates for elliptic problems [164]. It should be noted, however, that it yields formulas that are not necessarily locally consistent with the differential equations, if Taylor series expansions are substituted for the solution at the vertices appearing in the local stencil. Figure 7 illustrates the discretization of the Laplacian $u_{xx} + u_{yy}$ which is obtained with linear elements. It shows a particular triangulation such that the approximation is locally consistent with $u_{xx} + 3u_{yy}$. Thus the use of an irregular triangulation in the boundary layer may significantly degrade the accuracy.

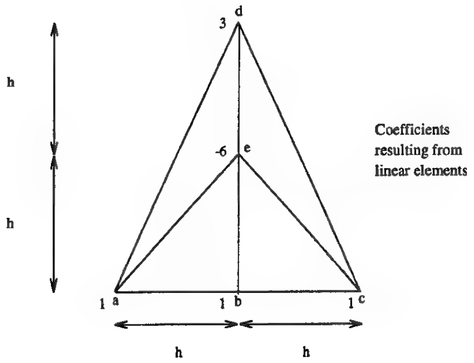


Figure 7: Example of discretization $u_{xx} + u_{yy}$ on a triangular mesh. The discretization is locally equivalent to the approximation $u_{xx}=\frac{u_a-2u_b+u_c}{h^2}$, $3u_{yy}=\frac{3u_d-6u_e+3u_b}{h^2}$.

4.7 Time Stepping Schemes

If the space discretization procedure is implemented separately, it leads to a set of coupled ordinary differential equations, which can be written in the form

$$\frac{d\mathbf{w}}{dt} + \mathbf{R}(\mathbf{w})=0, \qquad (40)$$

where $\mathbf{w}$ is the vector of the flow variables at the mesh points, and $\mathbf{R}(\mathbf{w})$ is the vector of the residuals, consisting of the flux balances defined by the space discretization scheme, together with the added dissipative terms. If the objective is simply to reach the steady state and details of the transient solution are immaterial, the time-stepping scheme may be designed solely to maximize the rate of convergence. The first decision that must be made is whether to use an explicit scheme, in which the space derivatives are calculated from known values of the flow variables at the beginning of the time step, or an implicit scheme, in which the formulas for the space derivatives include as yet unknown values of the flow variables at the end of the time step, leading to the need to solve coupled equations for the new values. The permissible time step for an explicit scheme is limited by the Courant-Friedrichs-Lewy (CFL) condition, which states that a difference scheme cannot be a convergent and stable approximation unless its domain of dependence contains the domain of dependence of the corresponding differential equation. One can anticipate that implicit schemes will yield convergence in a smaller number of time steps, because the time step is no longer constrained by the CFL condition. Implicit schemes will be efficient, however, only if the decrease in the number of time steps outweighs the increase in the computational effort per time step consequent upon the need to solve coupled equations. The prototype implicit scheme can be formulated by estimating $\frac{\partial \mathbf{w}}{\partial t}$ at $t + \mu\Delta t$ as a linear combination of $\mathbf{R}(\mathbf{w^n})$ and $\mathbf{R}(\mathbf{w^{n+1}})$. The resulting equation

$$\mathbf{w}^{n+1}=\mathbf{w}^n - \Delta t\left\{(1-\mu)\mathbf{R}\left(\mathbf{w^n}\right) + \mu\mathbf{R}\left(\mathbf{w}^{n+1}\right)\right\}$$

can be linearized as

$$\left(\mathbf{I} + \mu\Delta t\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\right)\delta\mathbf{w} + \Delta t\mathbf{R}(\mathbf{w}^n)=0.$$

If one sets μ=1 and lets $\Delta t \to \infty$ this reduces to the Newton iteration , which has been successfully used in two-dimensional calculations [173, 50]. In the three-dimensional case with, say, an $N \times N \times N$ mesh, the bandwidth of the matrix that must be inverted is of order N^2. Direct inversion requires a number of operations proportional to the number of unknowns multiplied by the square of the bandwidth of the order of N^7. This is prohibitive, and forces recourse to either an approximate factorization method or an iterative solution method.

Alternating direction methods, which introduce factors corresponding to each coordinate, are widely used for structured meshes [17, 137]. They cannot be implemented on unstructured tetrahedral meshes that do not contain identifiable mesh directions, although other decompositions are possible [108]. If one chooses to adopt the iterative solution technique, the principal alternatives are variants of the Gauss-Seidel and Jacobi methods. A symmetric Gauss-Seidel method with one iteration per time step is essentially equivalent to an approximate lower-upper (LU) factorization of the implicit scheme [86, 125, 31, 184]. On the other hand, the Jacobi method with a fixed number of iterations per time step reduces to a multistage explicit scheme, belonging to the general class of Runge-Kutta schemes [33]. Schemes of this type have proved very effective for wide variety of problems, and they have the advantage that they can be applied equally easily on both structured and unstructured meshes [84, 67, 69, 145].

If one reduces the linear model problem corresponding to (40) to an ordinary differential equation by substituting a Fourier mode $\hat{w}=e^{ipx_j}$, the resulting Fourier symbol has an imaginary part proportional to the wave speed, and a negative real part proportional to the diffusion. Thus the time stepping scheme should have a stability region which contains substantial intervals of both the negative real axis and the imaginary axis. To achieve this it pays to treat the convective and dissipative terms in a distinct fashion. Thus the residual is split as

$$R(w)=Q(w)+D(w),$$

where $Q(w)$ is the convective part and $D(w)$ the dissipative part. Denote the time level $n\Delta t$ by a superscript n. Then the multistage time stepping scheme is formulated as

$$\begin{aligned}
w^{(n+1,0)} &= w^n \\
&\dots \\
w^{(n+1,k)} &= w^n - \alpha_k\Delta t\left(Q^{(k-1)} + D^{(k-1)}\right) \\
&\dots \\
w^{n+1} &= w^{(n+1,m)},
\end{aligned}$$

where the superscript k denotes the k-th stage, α_m=1, and

$$\begin{aligned}
Q^{(0)} &= Q(w^n),\ D^{(0)}=D(w^n) \\
&\dots \\
Q^{(k)} &= Q\left(w^{(n+1,k)}\right) \\
D^{(k)} &= \beta_k D\left(w^{(n+1,k)}\right) + (1-\beta_k)D^{(k-1)}.
\end{aligned}$$

The coefficients α_k are chosen to maximize the stability interval along the imaginary axis, and the coefficients

β_k are chosen to increase the stability interval along the negative real axis.

These schemes do not fall within the standard framework of Runge-Kutta schemes, and they have much larger stability regions [69]. Two schemes which have been found to be particularly effective are tabulated below. The first is a four-stage scheme with two evaluations of dissipation. Its coefficients are

$$\begin{array}{ll} \alpha_1=\frac{1}{3} & \beta_1=1 \\ \alpha_2=\frac{4}{15} & \beta_2=\frac{1}{2} \\ \alpha_3=\frac{5}{9} & \beta_3=0 \\ \alpha_4=1 & \beta_4=0 \end{array} . \tag{41}$$

The second is a five-stage scheme with three evaluations of dissipation. Its coefficients are

$$\begin{array}{ll} \alpha_1=\frac{1}{4} & \beta_1=1 \\ \alpha_2=\frac{1}{6} & \beta_2=0 \\ \alpha_3=\frac{3}{8} & \beta_3=0.56 \\ \alpha_4=\frac{1}{2} & \beta_4=0 \\ \alpha_5=1 & \beta_5=0.44 \end{array} . \tag{42}$$

4.8 Multigrid Methods

4.8.1 Acceleration of Steady Flow Calculations

Radical improvements in the rate of convergence to a steady state can be realized by the multigrid time-stepping technique. The concept of acceleration by the introduction of multiple grids was first proposed by Fedorenko [48]. There is by now a fairly well-developed theory of multigrid methods for elliptic equations based on the concept that the updating scheme acts as a smoothing operator on each grid [24, 53]. This theory does not hold for hyperbolic systems. Nevertheless, it seems that it ought to be possible to accelerate the evolution of a hyperbolic system to a steady state by using large time steps on coarse grids so that disturbances will be more rapidly expelled through the outer boundary. Various multigrid time-stepping schemes designed to take advantage of this effect have been proposed [124, 65, 55, 71, 29, 6, 57, 83, 93].

One can devise a multigrid scheme using a sequence of independently generated coarser meshes by eliminating alternate points in each coordinate direction. In order to give a precise description of the multigrid scheme, subscripts may be used to indicate the grid. Several transfer operations need to be defined. First the solution vector on grid k must be initialized as

$$w_k^{(0)}=T_{k,k-1}w_{k-1},$$

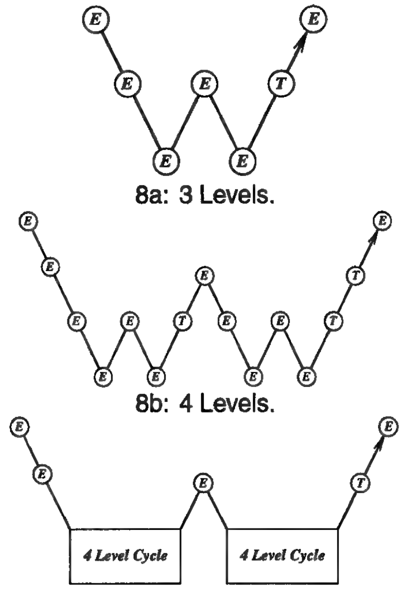
where w_{k-1} is the current value on grid $k-1$, and $T_{k,k-1}$ is a transfer operator. Next it is necessary to transfer a residual forcing function such that the solution grid k is driven by the residuals calculated on grid $k-1$. This can be accomplished by setting

$$P_k=Q_{k,k-1}R_{k-1}\left(w_{k-1}\right)-R_k\left[w_k^{(0)}\right],$$

where $Q_{k,k-1}$ is another transfer operator. Then $R_k\left(w_k\right)$ is replaced by $R_k\left(w_k\right)+P_k$ in the time-stepping scheme. Thus, the multistage scheme is reformulated as

$$\begin{array}{rcl} w_k^{(1)} & = & w_k^{(0)}-\alpha_1\Delta t_k\left[R_k^{(0)}+P_k\right] \\ \cdots & & \cdots \\ w_k^{(q+1)} & = & w_k^{(0)}-\alpha_{q+1}\Delta t_k\left[R_k^{(q)}+P_k\right]. \end{array}$$

The result $w_k^{(m)}$ then provides the initial data for grid $k+1$. Finally, the accumulated correction on grid k has to be transferred back to grid $k-1$ with the aid of an interpolation operator $I_{k-1,k}$. With properly optimized coefficients multistage time-stepping schemes can be very efficient drivers of the multigrid process. A W-cycle of the type illustrated in Figure 8 proves to be a particularly effective strategy for managing the work split between the meshes. In a three-dimensional case the number of cells is reduced by a factor of eight on each coarser grid. On examination of the figure, it can therefore be seen that the work measured in units corresponding to a step on the fine grid is of the order of



8a: 3 Levels.

8b: 4 Levels.

8c: 5 Levels.

Figure 8: Multigrid W-cycle for managing the grid calculation. E, evaluate the change in the flow for one step; T, transfer the data without updating the solution.

$$1+2/8+4/64+\ldots<4/3,$$

and consequently the very large effective time step of the complete cycle costs only slightly more than a single time step in the fine grid.

4.8.2 Multigrid Implicit Schemes for Unsteady Flow

Time dependent calculations are needed for a number of important applications, such as flutter analysis, or the analysis of the flow past a helicopter rotor, in which the stability limit of an explicit scheme forces the use of much smaller time steps than would be needed for an accurate simulation. In this situation a multigrid explicit scheme can be used in an inner iteration to solve the equations of a fully implicit time stepping scheme [74].

Suppose that (40) is approximated as

$$D_t w^{n+1}+R(w^{n+1})=0.$$

Here D_t is a k^{th} order accurate backward difference operator of the form

$$D_t=\frac{1}{\Delta t}\sum_{q=1}^{k}\frac{1}{q}\left(\Delta^-\right)^q,$$

where

$$\Delta^- w^{n+1}=w^{n+1}-w^n.$$

Applied to the linear differential equation

$$\frac{dw}{dt}=\alpha w$$

the schemes with k=1, 2 are stable for all $\alpha\Delta t$ in the left half plane (A-stable). Dahlquist has shown that A-stable linear multi-step schemes are at best second order accurate [38]. Gear however, has shown that the schemes with $k \le 6$ are stiffly stable [49], and one of the higher order schemes may offer a better compromise between accuracy and stability, depending on the application.

Equation (40) is now treated as a modified steady state problem to be solved by a multigrid scheme using variable local time steps in a fictitious time t^*. For example, in the case k=2 one solves

$$\frac{\partial w}{\partial t^*}=R^*(w),$$

where

$$R^*(w)=\frac{3}{2\Delta t}w+R(w)+\frac{2}{\Delta t}w^n-\frac{1}{2\Delta t}w^{n-1},$$

and the last two terms are treated as fixed source terms. The first term shifts the Fourier symbol of the equivalent model problem to the left in the complex plane. While this promotes stability, it may also require a limit to be imposed on the magnitude of the local time step Δt^* relative to that of the implicit time step Δt. This may be relieved by a point-implicit modification of the multistage scheme [119]. In the case of problems with moving boundaries the equations must be modified to allow for movement and deformation of the mesh.

This method has proved effective for the calculation of unsteady flows that might be associated with wing flutter [3, 4] and also in the calculation of unsteady incompressible flows [18]. It has the advantage that it can be added as an option to a computer program which uses an explicit multigrid scheme, allowing it to be used for the efficient calculation of both steady and unsteady flows.

4.9 Preconditioning

Another way to improve the rate of convergence to a steady state is to multiply the space derivatives in equation (1) by a preconditioning matrix P which is designed to equalize the eigenvalues, so that all the waves can be advanced with optimal time steps. A symmetric preconditioner which equalizes the eigenvalues has been proposed by Van Leer [102]. When the equations are written in stream-aligned coordinates this has the form

$$P=\begin{bmatrix} \frac{\tau}{\beta^2}M^2 & -\frac{\tau}{\beta}M & 0 & 0 & 0 \\ -\frac{\tau}{\beta}M & \frac{\tau}{\beta^2}+1 & 0 & 0 & 0 \\ 0 & 0 & \tau & 0 & 0 \\ 0 & 0 & 0 & \tau & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$$\begin{aligned} \beta &= \tau=\sqrt{1-M^2}, \quad \text{if} \quad M<1 \\ \beta &= \sqrt{1-M^2}, \tau=\sqrt{1-\frac{1}{M^2}}, \quad \text{if} \quad M \ge 1 \end{aligned}$$

Turkel has proposed an asymmetric preconditioner which has also proved effective, particularly for flow at low Mach numbers [172]. The use of these preconditioners can lead to instability at stagnation points where there is a zero eigenvalue which cannot be equalized with the eigenvalues $\pm c$.

The preconditioners of Van Leer and Turkel do not take account of the effect of differences in the mesh intervals in the different coordinate directions. The need to resolve the boundary layer generally forces the introduction of mesh cells with very high aspect ratios near the boundary, and these can lead to a severe reduction in the rate of convergence to a steady state. Pierce has recently obtained impressive results using diagonal and block-Jacobi preconditioners which include the mesh intervals [133].

An alternative approach has recently been proposed by Ta'asan [168], in which the equations are written in a canonical form which separates the equations describing acoustic waves from those describing convection. In terms of the velocity components u, v and the vorticity ω, temperature T, entropy s and total enthalpy H, the equations describing steady two-dimensional flow can be written as

$$\begin{bmatrix} D_1 & D_2 & 0 & 0 & 0 \\ -\frac{\partial}{\partial y} & \frac{\partial}{\partial x} & -1 & 0 & 0 \\ 0 & 0 & -q & -\frac{c^2}{\gamma(\gamma-1)}D_3 & \frac{1}{q}D_3 \\ 0 & 0 & 0 & T\rho Q & 0 \\ 0 & 0 & 0 & 0 & \rho Q \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \\ s \\ H \end{bmatrix}=0$$

where

$$\begin{aligned} D_1 &= \frac{\rho}{c^2}\left((c^2-u^2)\frac{\partial}{\partial x}-uv\frac{\partial}{\partial y}\right) \\ D_2 &= \frac{\rho}{c^2}\left((c^2-u^2)\frac{\partial}{\partial y}-uv\frac{\partial}{\partial x}\right) \\ D_3 &= v\frac{\partial}{\partial x}-u\frac{\partial}{\partial y} \\ Q &= u\frac{\partial}{\partial x}+v\frac{\partial}{\partial y} \end{aligned}$$

Here the first two equations describe an elliptic system if the flow is subsonic, while the remaining equations are convective. Now separately optimized multigrid procedures are used to solve the two sets of equations, which are essentially decoupled.

4.10 High Order Schemes and Mesh Refinement

The need both to improve the accuracy of computational simulations and to assure known levels of accuracy is the focus of ongoing research. The main routes to improving the accuracy are to increase the order of the discrete scheme and to reduce the mesh interval. High order difference methods are most easily implemented on Cartesian, or at least extremely smooth grids. The expansion of the stencil as the order is increased leads to the need for complex boundary conditions. Compact schemes keep the stencil as small as possible [140, 104, 28]. On simple domains, spectral methods are particularly effective, especially in the case of periodic boundary conditions, and can be used to produce exponentially fast convergence of the error as the mesh interval is decreased [127, 27]. A compromise is to divide the field into subdomains and introduce high order elements. This approach is used in the spectral element method [92].

High order difference schemes and spectral methods have proven particularly useful in direct Navier-Stokes simulations of transient and turbulent flows. High order methods are also beneficial in computational aero-acoustics, where it is desired to track waves over long distances with minimum error. If the flow contains shock waves or contact discontinuities, the ENO method may be used to construct high order non-oscillatory schemes.

In multi-dimensional flow simulations, global reduction of the mesh interval can be prohibitively expensive, motivating the use of adaptive mesh refinement procedures which reduce the local mesh width h if there is an indication that the error is too large [21, 39, 109, 61, 138, 103]. In such h-refinement methods, simple error indicators such as local solution gradients may be used. Alternatively, the discretization error may be estimated by comparing quantities calculated with two mesh widths, say on the current mesh and a coarser mesh with double the mesh interval. Procedures of this kind may also be used to provide *a posteriori* estimates of the error once the calculation is completed.

This kind of local adaptive control can also be applied to the local order of a finite element method to produce a p-refinement method, where p represents the order of the polynomial basis functions. Finally, both h- and p-refinement can be combined to produce an h-p method in which h and p are locally optimized to yield a solution with minimum error [126]. Such methods can achieve exponentially fast convergence, and are well established in computational solid mechanics.

5. CURRENT STATUS OF NUMERICAL SIMULATION

This section presents some representative numerical results which confirm the properties of the algorithms which have been reviewed in the last section. These have been drawn from the work of the author and his associates. They also illustrate the kind of calculation which can be performed in an industrial environment, where rapid turn around is important to allow the quick assessment of design changes, and computational costs must be limited.

5.1 One-dimensional shock

In order to verify the discrete structure of stationary shocks, calculations were performed for a one-dimensional problem with initial data containing left and right states compatible with the Rankine Hugoniot conditions. An intermediate state consisting of the arithmetic average of the left and right states was introduced at a single cell in the center of the domain. With this intermediate state the system is not in equilibrium, and the time dependent equations were solved to find an equilibrium solution with a stationary shock wave separating the left and right states. Table 1 shows the result for a shock wave at Mach 20. This calculation used the H-CUSP scheme, which allows a solution with constant stagnation enthalpy, with the limiter defined by equation (23), and q=3. The formulation is described in detail in reference [80]. The table shows the values of H, p, M and the entropy $S=\log \frac{p}{\rho\gamma} - \log\left(\frac{p_L}{\rho_L^\gamma}\right)$. A perfect one point shock structure is displayed. The entropy is zero to 4 decimal places upstream of the shock, exhibits a slight excursion at the interior point, and is constant to 4 decimal places downstream of the shock. It may be noted that the mass, momentum and energy of the initial data are not compatible with the final equilibrium state. According to conservation arguments the total mass, momentum and energy must remain constant if the outflow flux f_R remains equal to the inflow flux f_L. Therefore f_R must be allowed to vary according to an appropriate outflow boundary condition to allow the total mass, momentum and energy to be adjusted to values compatible with equilibrium.

I	H	p	M	s
19	283.5000	1.0000	20.0000	0.0000
20	283.5000	1.0000	20.0000	0.0000
21	283.5000	1.0000	20.0000	0.0000
22	283.4960	307.4467	0.7229	40.3353
23	283.4960	466.4889	0.3804	37.6355
24	283.4960	466.4889	0.3804	37.6355
25	283.4960	466.4889	0.3804	37.6355

Table 1: Shock Wave at Mach 20

5.2 Euler Calculations for Airfoils and Wings

The results of transonic flow calculations for two well known airfoils, the RAE 2822 and the NACA 0012, are presented in figures (22-25). The H-CUSP scheme was again used. The limiter defined by equation (23) was used with q=3. The 5 stage time stepping scheme (42) was augmented by the multigrid scheme described in section 4.2 to accelerate convergence to a steady state. The equations were discretized on meshes with O-topology extending out to a radius of about 100 chords. In each case the calculations were performed on a sequence of successively finer meshes from 40x8 to 320x64 cells, while the multigrid cycles on each of these meshes descended to a coarsest mesh of 10x2 cells. Figure 22 shows the inner parts of the 160x32 meshes for the two airfoils. Figures 23-25 show the final results on 320x64 meshes for the RAE 2822 airfoil at Mach .75 and 3° angle of attack, and for the NACA 0012 airfoil at Mach .8 and 1.25° angle of attack, and also at Mach .85 and 1° angle of attack. In the pressure distributions the pressure coefficient $C_p=\frac{p-p_\infty}{\frac{1}{2}\rho_\infty q_\infty^2}$ is plotted with the negative (suction) pressures upward, so that the upper curve represents the flow over the upper side of a lifting airfoil. The convergence histories show the mean rate of change of the density, and also the total number of supersonic points in the flow field, which provides a useful measure of the global convergence of transonic flow calculations such as these. In each case the convergence history is shown for 100 cycles, while the pressure distribution is displayed after a sufficient number of cycles for its convergence. The pressure distribution of the RAE 2822 airfoil converged in only 25 cycles. Convergence was slower for the NACA 0012 airfoil. In the case of flow at Mach .8 and 1.25° angle of attack, additional cycles were needed to damp out a wave downstream of the weak shock wave on the lower surface.

As a further check on accuracy the drag coefficient should be zero in subsonic flow, or in shock free transonic flow. Table 2 shows the computed drag coefficient on a sequence of three meshes for three examples. The first two are subsonic flows over the RAE 2822 and NACA 0012 airfoils at Mach .5 and 3° angle of attack. The third is the flow over the shock free Korn airfoil at its design point of Mach .75 and 0° angle of attack. In all three cases the drag coefficient is calculated to be zero to four digits on a 160x32 mesh.

Mesh	RAE 2822	NACA 0012	Korn Airfoil
	Mach .50	Mach .50	Mach .75
	α 3°	α 3°	α 0°
40x8	.0062	.0047	.0098
80x16	.0013	.0008	.0017
160x32	.0000	.0000	.0000

Table 2: Drag Coefficient on a sequence of meshes

As a further test of the performance of the H-CUSP scheme, the flow past the ONERA M6 wing was calculated on a mesh with C-H topology and 192x32x48 = 294912 cells. Figure 26 shows the result at Mach .84 and 3.06° angle of attack. This again verifies the non-oscillatory character of the solution, and the sharp resolution of shock waves. In this case 50 cycles were sufficient for convergence of the pressure distributions.

Figure 9 shows a calculation of the Northrop YF23 by R.J. Busch, Jr., who used the author's FLO57 code to solve the Euler equations [26]. Although an inviscid model of the flow was used, it can be seen that the simulations are in good agreement with wind tunnel measurements both at Mach .90, with angles of attack of 0, 8 and 16 degrees, and at Mach 1.5 with angles of attack of 0, 4 and 8 degrees. At a high angle of attack the flow separates from the leading edge, and this example shows that in situations where the point of separation is fixed, an inviscid model may still produce a useful prediction. Thus valuable information for the aerodynamic design could be obtained with a relatively inexpensive computational model.

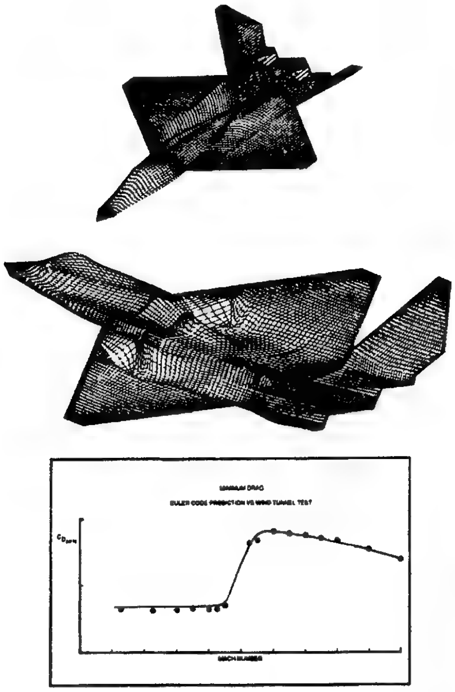


Figure 9: Comparison of Experimental and Computed Drag Rise Curve for the YF-23 (Supplied by R. J. Bush Jr.)

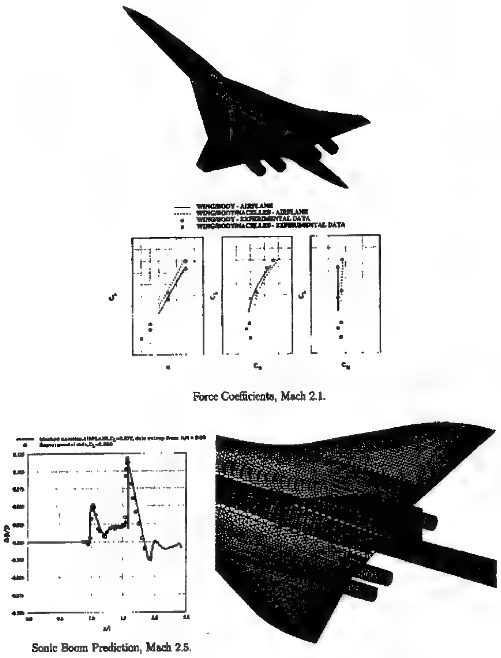


Figure 10: Comparison of Experimental and Calculated Results for a HSCT Configuration

The next figures show the results of calculations using the AIRPLANE code developed by T.J. Baker and the author, to solve the Euler equations on an unstructured mesh. This provides the flexibility to treat arbitrarily complex configurations without the need to spend months developing an acceptable mesh. Figures 10 and 11 show calculations for supersonic transport configurations which were performed by Susan Cliff. The agreement with experimental data is quite good, and it has also been possible to predict the sonic boom signature [34]. Figure 12 shows an Euler calculation for the McDonnell Douglas MD11 with flow through the engine nacelles, using 348407 mesh points of 2100466 tetrahedra. This calculation takes 4 hours on an IBM 590 workstation. A parallel version of the code has been developed in collaboration with W.S. Cheng, and the same calculation can be performed in 20 minutes using 16 processors of an IBM SP2. The parallel speed-up for the MD11 is shown in table 3.

No. of Nodes	Seconds/Cycle	Speedup
1	36.03	1.00
2	18.11	1.99
4	9.11	3.96
8	4.66	7.73
16	2.39	15.08

Table 3: AIRPLANE Parallel Performance on the SP2, MD-11 Model

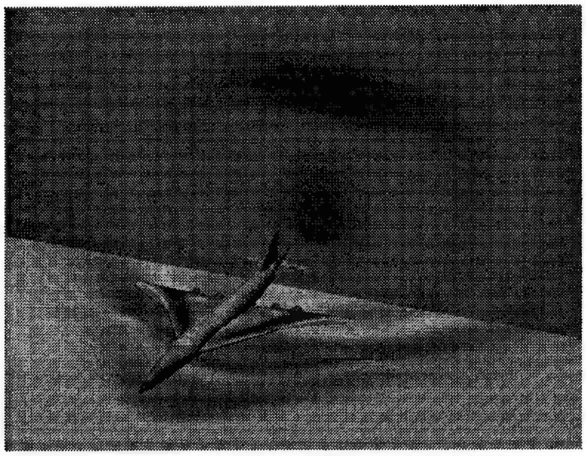


Figure 11: Pressure Contours and Sonic Boom on a Representative HSCT Configuration

5.3 Viscous Flow Calculations

The next figures show viscous simulations based on the solution of the Reynolds averaged Navier Stokes equations with turbulence models. Figure 13 shows a two-dimensional calculation for the RAE 2822 airfoil by L. Martinelli. The vertical axis represents the negative pressure coefficient, and there is a shock wave half way along the upper surface. This example confirms that in the absence of significant shock induced separation, simulations performed on a sufficiently fine mesh (with 512 × 64 cells) can produce excellent agreement with experimental data. Figure 21 shows a simulation of the McDonnell-Douglas F18 performed by R.M. Cummings, Y.M. Rizk, L.B. Schiff and N.M. Chaderjian at NASA Ames [37]. They used a multiblock mesh with about 900000 mesh points. While this is probably not enough for an accurate quantitative prediction, the agreement with both the experimental data and the visualization are quite good.

Figure 14 shows an unsteady flow calculation for a pitching airfoil performed by J. Alonso using the code UFLO82, which he jointly developed with L. Martinelli and the author [4]. This uses the multigrid implicit scheme described in Section 3.7.2 which allows the number of time steps to be reduced from several thousand to 36 per pitching cycle. The agreement with experimental data is quite good.

Figure 12: Computed Pressure Field for a McDonnell Douglas MD11

5.4 Ship Wave Resistance calculations

Figures 15-17 show the results of an application of the same multigrid finite volume techniques to the calculation of the flow past a naval frigate, using a code which was developed by J. Farmer, L. Martinelli and the author [47]. The mesh was adjusted during the course of the calculation to conform to the free surface in order to satisfy the exact non-linear boundary condition, while artificial compressibility was used to treat the incompressible flow equations.

6. AERODYNAMIC SHAPE OPTIMIZATION

6.1 Optimization and Design

Traditionally the process of selecting design variations has been carried out by trial and error, relying on the intuition and experience of the designer. With currently available equipment the turn around for numerical simulations is becoming so rapid that it is feasible to examine an extremely large number of variations. It is not at all likely that repeated trials in an interactive design and analysis procedure can lead to a truly optimum design. In order to take full advantage of the possibility of examining a large design space the numerical simulations need to be combined with automatic search and optimization procedures. This can lead to automatic design methods which will fully realize the potential improvements in aerodynamic efficiency.

The simplest approach to optimization is to define the geometry through a set of design parameters, which may, for example, be the weights α_i applied to a set of shape functions $b_i(x)$ so that the shape is represented as

$$f(x) = \sum \alpha_i b_i(x) .$$

Then a cost function I is selected which might, for example, be the drag coefficient or the lift to drag ratio, and I is regarded as a function of the parameters α_i. The sensitivities $\frac{\partial I}{\partial \alpha_i}$ may now be estimated by making a small variation $\delta\alpha_i$ in each design parameter in turn and recalculating the flow to obtain the change in I. Then

$$\frac{\partial I}{\partial \alpha_i} \approx \frac{I(\alpha_i + \delta\alpha_i) - I(\alpha_i)}{\delta\alpha_i}.$$

The gradient vector $\frac{\partial I}{\partial \alpha}$ may now be used to determine a direction of improvement. The simplest procedure is to make a step in the negative gradient direction by setting

$$\alpha^{n+1} = \alpha^n - \lambda\delta\alpha,$$

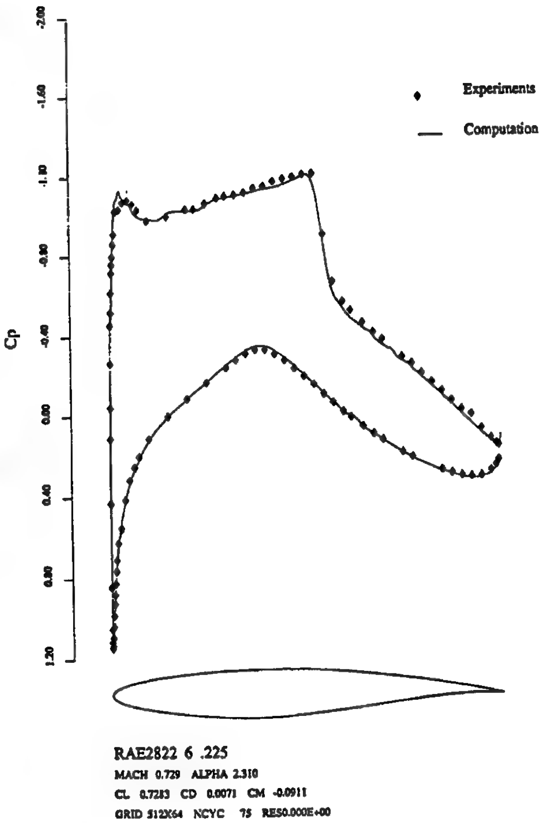


Figure 13: Two-Dimensional Turbulent Viscous Calculation (by Luigi Martinelli)

so that to first order

$$I + \delta I = I - \frac{\partial I^T}{\partial \alpha}\delta\alpha = I - \lambda\frac{\partial I^T}{\partial \alpha}\frac{\partial I}{\partial \alpha}.$$

More sophisticated search procedures may be used such as quasi-Newton methods, which attempt to estimate the second derivative $\frac{\partial^2 I}{\partial \alpha_i \partial \alpha_j}$ of the cost function from changes in the gradient $\frac{\partial I}{\partial \alpha}$ in successive optimization steps. These methods also generally introduce line searches to find the minimum in the search direction which is defined at each step. The main disadvantage of this approach is the need for a number of flow calculations proportional to the number of design variables to estimate the gradient. The computational costs can thus become prohibitive as the number of design variables is increased.

An alternative approach is to cast the design problem as a search for the shape that will generate the desired pressure distribution. This approach recognizes that the designer usually has an idea of the the kind of pressure distribution that will lead to the desired performance. Thus, it is useful to consider the inverse problem of calculating the shape that will lead to a given pressure distribution. The method has the advantage that only one flow solution is required to obtain the desired design. Unfortunately, a physically realizable shape may not necessarily exist, unless the pressure distribution satisfies certain constraints. Thus the problem must be very carefully formulated; otherwise it may be ill posed.

The difficulty that the target pressure may be unattainable may be circumvented by treating the inverse problem as a special case of the optimization problem, with a cost function which measures the error in the solution of the inverse problem. For example, if p_d is the desired surface pressure, one may take the cost function to be an integral over the the body surface of the square of the pressure

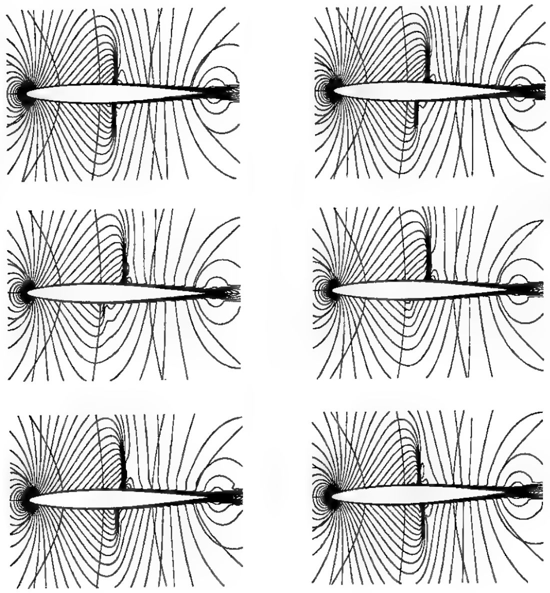


Figure 14: Mach Number Contours. Pitching Airfoil Case. Re=1.0 × 10^6, M_∞=0.796, K_c=0.202.

error,

$$I=\frac{1}{2}\int_{\mathcal{B}}(p-p_d)^2\,d\mathcal{B},$$

or possibly a more general Sobolev norm of the pressure error. This has the advantage of converting a possibly ill posed problem into a well posed one. It has the disadvantage that it incurs the computational costs associated with optimization procedures.

6.2 Application of Control Theory

In order to reduce the computational costs, it turns out that there are advantages in formulating both the inverse problem and more general aerodynamic problems within the framework of the mathematical theory for the control of systems governed by partial differential equations [105]. A wing, for example, is a device to produce lift by controlling the flow, and its design can be regarded as a problem in the optimal control of the flow equations by variation of the shape of the boundary. If the boundary shape is regarded as arbitrary within some requirements of smoothness, then the full generality of shapes cannot be defined with a finite number of parameters, and one must use the concept of the Frechet derivative of the cost with respect to a function. Clearly, such a derivative cannot be determined directly by finite differences of the design parameters because there are now an infinite number of these. Using techniques of control theory, however, the gradient can be determined indirectly by solving an adjoint equation which has coefficients defined by the solution of the flow equations. The cost of solving the adjoint equation is comparable to that of solving the flow equations. Thus the gradient can be determined with roughly the computational costs of two flow solutions, independently of the number of design variables, which may be infinite if the boundary is regarded as a free surface.

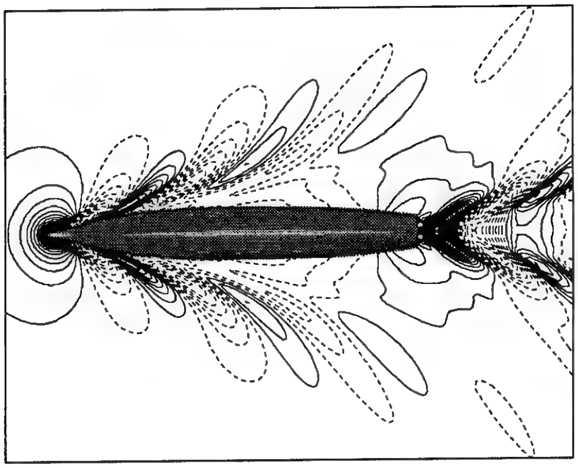


Figure 15: Contours of Surface Wave Elevation for a Combatant Ship

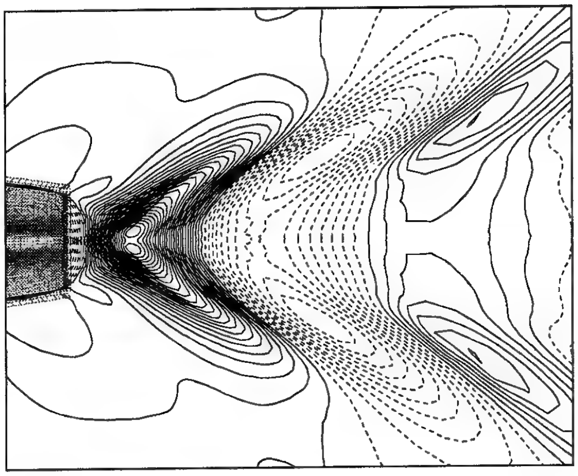


Figure 16: Contours of Surface Wave Elevation Near the Transom Stern

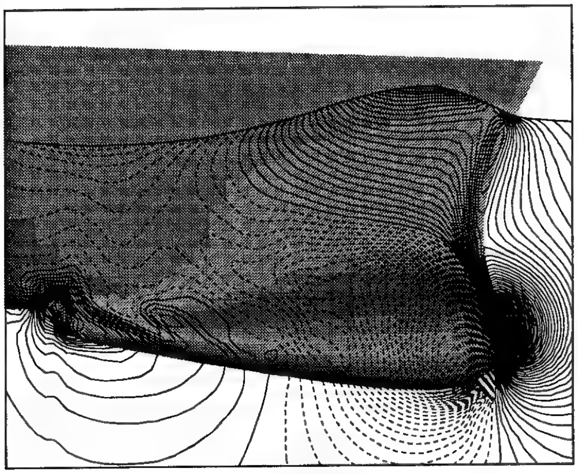


Figure 17: Pressure Contours in the Bow Region

For flow about an airfoil or wing, the aerodynamic properties which define the cost function are functions of the flow-field variables (w) and the physical location of the boundary, which may be represented by the function $\mathcal{F}$, say. Then

$$I=I\left(w,\mathcal{F}\right),$$

and a change in $\mathcal{F}$ results in a change

$$\delta I=\frac{\partial I^T}{\partial w}\delta w+\frac{\partial I^T}{\partial \mathcal{F}}\delta\mathcal{F}, \qquad (43)$$

in the cost function. Using control theory, the governing equations of the flowfield are introduced as a constraint in such a way that the final expression for the gradient does not require reevaluation of the flowfield. In order to achieve this δw must be eliminated from (43). Suppose that the governing equation R which expresses the dependence of w and $\mathcal{F}$ within the flowfield domain D can be written as

$$R(w,\mathcal{F})=0. \tag{44}$$

Then δw is determined from the equation

$$\delta R=\left[\frac{\partial R}{\partial w}\right]\delta w+\left[\frac{\partial R}{\partial \mathcal{F}}\right]\delta \mathcal{F}=0. \tag{45}$$

Next, introducing a Lagrange Multiplier ψ, we have

$$\begin{aligned}\delta I &= \frac{\partial I^T}{\partial w}\delta w+\frac{\partial I^T}{\partial \mathcal{F}}\delta \mathcal{F}-\psi^T\left(\left[\frac{\partial R}{\partial w}\right]\delta w+\left[\frac{\partial R}{\partial \mathcal{F}}\right]\delta \mathcal{F}\right)\\ &= \left\{\frac{\partial I^T}{\partial w}-\psi^T\left[\frac{\partial R}{\partial w}\right]\right\}\delta w+\left\{\frac{\partial I^T}{\partial \mathcal{F}}-\psi^T\left[\frac{\partial R}{\partial \mathcal{F}}\right]\right\}\delta \mathcal{F}.\end{aligned}$$

Choosing ψ to satisfy the adjoint equation

$$\left[\frac{\partial R}{\partial w}\right]^T\psi=\frac{\partial I}{\partial w} \tag{46}$$

the first term is eliminated, and we find that

$$\delta I=\mathcal{G}\delta \mathcal{F}, \tag{47}$$

where

$$\mathcal{G}=\frac{\partial I^T}{\partial \mathcal{F}}-\psi^T\left[\frac{\partial R}{\partial \mathcal{F}}\right]$$

The advantage is that (47) is independent of δw, with the result that the gradient of I with respect to an arbitrary number of design variables can be determined without the need for additional flow-field evaluations. In the case that (44) is a partial differential equation, the adjoint equation (46) is also a partial differential equation and appropriate boundary conditions must be determined.

After making a step in the negative gradient direction, the gradient can be recalculated and the process repeated to follow a path of steepest descent until a minimum is reached. In order to avoid violating constraints, such as a minimum acceptable wing thickness, the gradient may be projected into the allowable subspace within which the constraints are satisfied. In this way one can devise procedures which must necessarily converge at least to a local minimum, and which can be accelerated by the use of more sophisticated descent methods such as conjugate gradient or quasi-Newton algorithms. There is the possibility of more than one local minimum, but in any case the method will lead to an improvement over the original design. Furthermore, unlike the traditional inverse algorithms, any measure of performance can be used as the cost function.

In reference [72] the author derived the adjoint equations for transonic flows modelled by both the potential flow equation and the Euler equations. The theory was developed in terms of partial differential equations, leading to an adjoint partial differential equation. In order to obtain numerical solutions both the flow and the adjoint equations must be discretized. The control theory might be applied directly to the discrete flow equations which result from the numerical approximation of the flow equations by finite element, finite volume or finite difference procedures. This leads directly to a set of discrete adjoint equations with a matrix which is the transpose of the Jacobian matrix of the full set of discrete nonlinear flow equations. On a three-dimensional mesh with indices i, j, k the individual adjoint equations may be derived by collecting together all the terms multiplied by the variation $\delta w_{i,j,k}$ of the discrete flow variable $w_{i,j,k}$. The resulting discrete adjoint equations represent a possible discretization of the adjoint partial differential equation. If these equations are solved exactly they can provide an exact gradient of the inexact cost function which results from the discretization of the flow equations. On the other hand any consistent discretization of the adjoint partial differential equation will yield the exact gradient in the limit as the mesh is refined. The trade-off between the complexity of the adjoint discretization, the accuracy of the resulting estimate of the gradient, and its impact on the computational cost to approach an optimum solution is a subject of ongoing research.

The true optimum shape belongs to an infinitely dimensional space of design parameters. One motivation for developing the theory for the partial differential equations of the flow is to provide an indication in principle of how such a solution could be approached if sufficient computational resources were available. Another motivation is that it highlights the possibility of generating ill posed formulations of the problem. For example, if one attempts to calculate the sensitivity of the pressure at a particular location to changes in the boundary shape, there is the possibility that a shape modification could cause a shock wave to pass over that location. Then the sensitivity could become unbounded. The movement of the shock, however, is continuous as the shape changes. Therefore a quantity such as the drag coefficient, which is determined by integrating the pressure over the surface, also depends continuously on the shape. The adjoint equation allows the sensitivity of the drag coefficient to be determined without the explicit evaluation of pressure sensitivities which would be ill posed.

The discrete adjoint equations, whether they are derived directly or by discretization of the adjoint partial differential equation, are linear. Therefore they could be solved by direct numerical inversion. The cost of direct inversion can become prohibitive, however, as the mesh is refined, and it becomes more efficient to use iterative solution methods. Moreover, because of the similarity of the adjoint equations to the flow equations, the same iterative methods which have been proved to be efficient for the solution of the flow equations are efficient for the solution of the adjoint equations.

The control theory formulation for optimal aerodynamic design has proved effective in a variety of applications [73, 77, 142]. The adjoint equations have also been used by Ta'asan, Kuruvila and Salas [167], who have implemented a one shot approach in which the constraint represented by the flow equations is only required to be satisfied by the final converged solution, and computational costs are also reduced by applying multigrid techniques to the geometry modifications as well as the solution of the flow and adjoint equations. Pironneau has studied the use of control theory for optimal shape design of systems governed by elliptic equations [134], and more recently the Navier-Stokes equations, and also wave reflection problems. Adjoint methods have also been used by Baysal and Eleshaky [16].

6.3 Three-Dimensional Design using the Euler Equations

In order to illustrate the application of control theory to aerodynamic design problems, this section treats the case of three-dimensional wing design using the inviscid Euler equations as the mathematical model for compressible flow. A transformation to a body-fitted coordinate system will be introduced, so that variations in the wing shape induce corresponding variations in the computational mesh. Thus the flow is determined by the solution of the transformed equation (5). Let

$$K_{ij}=\left[\frac{\partial x_i}{\partial \xi_j}\right],\ J=\det(K),\ K_{ij}^{-1}=\left[\frac{\partial \xi_i}{\partial x_j}\right],$$

and

$$Q=JK^{-1}.$$

The elements of Q are the coefficients of K, and in a finite volume discretization they are just the face areas of the computational cells projected in the x_1, x_2, and x_3 directions. Also introduce scaled contravariant velocity components

$$U_i{=}Q_{ij}u_j.$$

The transformed equations can now be written as

$$\frac{\partial W}{\partial t}+\frac{\partial F_i}{\partial \xi_i}{=}0 \tag{48}$$

where

$$W{=}Jw$$

and

$$F_i{=}Q_{ij}f_j{=}\left[\begin{array}{c} \rho U_i \\ \rho U_i u_1 + Q_{i1}p \\ \rho U_i u_2 + Q_{i2}p \\ \rho U_i u_3 + Q_{i3}p \\ \rho U_i H \end{array}\right]$$

Assume now that the new computational coordinate system conforms to the wing in such a way that the wing surface B_W is represented by $\xi_2{=}0$. Then the flow is determined as the steady state solution of equation (48) subject to the flow tangency condition

$$U_2{=}0 \quad \text{on } B_W. \tag{49}$$

At the far field boundary B_F, conditions are specified for incoming waves, as in the two-dimensional case, while outgoing waves are determined by the solution.

The weak form of the Euler equations for steady flow can be written as

$$\int_{\mathcal{D}} \frac{\partial \phi^T}{\partial \xi_i} F_i d\mathcal{D}{=}\int_{\mathcal{B}} n_i \phi^T F_i d\mathcal{B}, \tag{50}$$

where the test vector ϕ is an arbitrary differentiable function and n_i is the outward normal at the boundary. If a differentiable solution w is obtained to this equation, it can be integrated by parts to give

$$\int_{\mathcal{D}} \phi^T \frac{\partial F_i}{\partial \xi_i} d\mathcal{D}{=}0$$

and since this is true for any ϕ, the differential form can be recovered. If the solution is discontinuous, equation (50) may be integrated by parts separately on either side of the discontinuity to recover the shock jump conditions.

Suppose now that it is desired to control the surface pressure by varying the wing shape. It is convenient to retain a fixed computational domain. Variations in the shape then result in corresponding variations in the mapping derivatives defined by K. Introduce the cost function

$$I{=}\frac{1}{2}\iint_{B_W} (p-p_d)^2 \, d\xi_1 d\xi_3,$$

where p_d is the desired pressure. The design problem is now treated as a control problem where the control function is the wing shape, which is to be chosen to minimize I subject to the constraints defined by the flow equations (48–50). A variation in the shape will cause a variation δp in the pressure and consequently a variation in the cost function

$$\delta I{=}\iint_{B_W} (p-p_d)\,\delta p \;\; d\xi_1 d\xi_3. \tag{51}$$

Since p depends on w through the equation of state (2), the variation δp can be determined from the variation δw. Define the Jacobian matrices

$$A_i{=}\frac{\partial f_i}{\partial w}, \quad C_i{=}Q_{ij}A_j. \tag{52}$$

The weak form of the equation for δw in the steady state becomes

$$\int_{\mathcal{D}} \frac{\partial \phi^T}{\partial \xi_i} \delta F_i d\mathcal{D}{=}\int_{\mathcal{B}} (n_i \phi^T \delta F_i)\; d\mathcal{B},$$

where

$$\delta F_i{=}C_i \delta w + \delta Q_{ij} f_j,$$

which should hold for any differential test function ϕ. This equation may be added to the variation in the cost function, which may now be written as

$$\begin{aligned} \delta I{=}&\iint_{B_W} (p-p_d)\,\delta p \;\; d\xi_1 d\xi_3 \\ &-\int_{\mathcal{D}} \left(\frac{\partial \psi^T}{\partial \xi_i}\delta F_i\right) d\mathcal{D} \\ &+\int_{\mathcal{B}} \left(n_i \psi^T \delta F_i\right) d\mathcal{B}. \end{aligned} \tag{53}$$

On the wing surface B_W, $n_1{=}n_3{=}0$ and it follows from equation (49) that

$$\delta F_2{=}\left[\begin{array}{c} 0 \\ Q_{21}\delta p \\ Q_{22}\delta p \\ Q_{23}\delta p \\ 0 \end{array}\right]+\left[\begin{array}{c} 0 \\ \delta Q_{21}p \\ \delta Q_{22}p \\ \delta Q_{23}p \\ 0 \end{array}\right] \tag{54}$$

Since the weak equation for δw should hold for an arbitrary choice of the test vector ϕ, we are free to choose ϕ to simplify the resulting expressions. Therefore we set $\phi{=}\psi$, where the costate vector ψ is the solution of the adjoint equation

$$\frac{\partial \psi}{\partial t}-C_i^T\frac{\partial \psi}{\partial \xi_i}{=}0 \quad \text{in } D. \tag{55}$$

At the outer boundary incoming characteristics for ψ correspond to outgoing characteristics for δw. Consequently one can choose boundary conditions for ψ such that

$$n_i \psi^T C_i \delta w{=}0.$$

Then if the coordinate transformation is such that δQ is negligible in the far field, the only remaining boundary term is

$$-\iint_{B_W} \psi^T \delta F_2 \; d\xi_1 d\xi_3.$$

Thus by letting ψ satisfy the boundary condition,

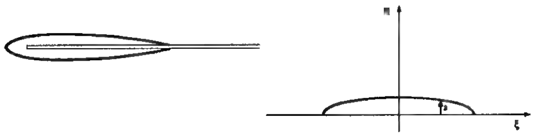
$$Q_{21}\psi_2 + Q_{22}\psi_3 + Q_{23}\psi_4{=}(p-p_d) \quad \text{on } B_W, \tag{56}$$

we find finally that

$$\begin{aligned} \delta I \quad {=}&-\int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_i}\delta Q_{ij} f_j d\mathcal{D} \\ &-\iint_{B_W}(\delta Q_{21}\psi_2 + \delta Q_{22}\psi_3 + Q_{23}\psi_4)\, p \, d\xi_1 d\xi_3. \end{aligned} \tag{57}$$

A convenient way to treat a wing is to introduce sheared parabolic coordinates as shown in figure 18 through the transformation

$$\begin{aligned} x &= x_0(\zeta)+\frac{1}{2}a(\zeta)\left\{\xi^2-(\eta+\mathcal{S}(\xi,\zeta))^2\right\} \\ y &= y_0(\zeta)+a(\zeta)\,\xi\,(\eta+\mathcal{S}(\xi,\zeta)) \\ z &= \zeta. \end{aligned}$$



18a: x, y-Plane. 18b: ξ, η-Plane.

Figure 18: Sheared Parabolic Mapping.

Here x=x_1, y=x_2, z=x_3 are the Cartesian coordinates, and ξ and $\eta + \mathcal{S}$ correspond to parabolic coordinates generated by the mapping

$$x + iy{=}x_0 + iy_0 + \frac{1}{2}a\left(\zeta\right)\left\{\xi + i\left(\eta + \mathcal{S}\right)\right\}^2$$

at a fixed span station ζ. $x_0\left(\zeta\right)$ and $y_0\left(\zeta\right)$ are the coordinates of a singular line which is swept to lie just inside the leading edge of a swept wing, while $a\left(\zeta\right)$ is a scale factor to allow for spanwise chord variations.

We now treat $\mathcal{S}\left(\xi, \zeta\right)$ as the control. Substitution of these formulas yields the variation in the form

$$\delta I{=}\int\int \mathcal{G}\left(\xi, \eta\right)\,\delta\mathcal{S}\left(\xi, \eta\right)\,d\xi d\eta$$

where the gradient $\mathcal{G}\left(\xi, \eta\right)$ is obtained by evaluating the integrals in equation (57). Thus to reduce I we can choose

$$\delta\mathcal{S}{=}-\lambda\mathcal{G}$$

where λ is sufficiently small and non-negative. In order to impose a thickness constraint we can define a baseline surface $\mathcal{S}_0\left(\xi, \zeta\right)$ below which $\mathcal{S}\left(\xi, \zeta\right)$ is not allowed to fall. Now we take λ=$\lambda\left(\xi, \zeta\right)$ as a non-negative function such that

$$\mathcal{S}\left(\xi, \zeta\right) + \delta\mathcal{S}\left(\xi, \zeta\right) \geq \mathcal{S}_0\left(\xi, \zeta\right). \qquad (58)$$

Then the constraint is satisfied, while

$$\delta I{=}-\int\int_{Bw} \lambda\mathcal{G}^2 d\xi\; d\zeta \leq 0.$$

The costate solution ψ is a legitimate test function for the weak form of the flow equations only if it is differentiable. Smoothness should also be preserved in the redesigned shape. It is therefore crucially important to introduce appropriate smoothing procedures. In order to avoid discontinuities in the adjoint boundary condition which would be caused by the appearance of shock waves, the cost function for the target pressure may be modified to the form

$$\begin{aligned} I &= \frac{1}{2}\int\int\left(\lambda_1\mathcal{Z} + \lambda_2\left(\frac{\partial\mathcal{Z}}{\partial\xi}\right)^2\right)d\xi d\eta \\ \lambda_1\mathcal{Z} &- \frac{\partial}{\partial\xi}\lambda_2\frac{\partial\mathcal{Z}}{\partial\xi}{=}p - p_d. \end{aligned}$$

Then

$$\begin{aligned} \delta I &= \int\int\left(\lambda_1\mathcal{Z}\delta\mathcal{Z} + \lambda_2\frac{\partial\mathcal{Z}}{\partial\xi}\frac{\partial}{\partial\xi}\delta\mathcal{Z}\right)d\xi d\eta \\ &= \int\int\mathcal{Z}\left(\lambda_1 - \frac{\partial}{\partial\xi}\lambda_2\frac{\partial}{\partial\xi}\right)\delta\mathcal{Z}\,d\xi d\eta \\ &= \int\int\mathcal{Z}\delta p\,d\xi d\eta \end{aligned}$$

and the smooth quantity $\mathcal{Z}$ replaces $p - p_d$ in the adjoint boundary condition.

Independent movement of the boundary mesh points could produce discontinuities in the designed shape. In order to prevent this the gradient may be also smoothed. Both explicit and implicit smoothing procedures are useful. Suppose that the movement of the surface mesh points were defined by local B-splines. In the case of a uniform one-dimensional mesh, a B-spline with a displacement d centered at the mesh point i would produce displacements $d/4$ at $i + 1$ and $i - 1$ and zero elsewhere, while preserving continuity of the first and second derivatives. Thus we can suppose that the discrete surface displacement has the form

$$\delta\mathcal{S}{=}Bd,$$

where B is a matrix with coefficients defined by the B-splines, and d_i is the displacement associated with the B-spline centered at i. Then, using the discrete formulas, to first order the change in the cost is

$$\delta I{=}\mathcal{G}^T\delta\mathcal{S}{=}\mathcal{G}^T Bd.$$

Thus the gradient with respect to the B-spline coefficients is obtained by multiplying $\mathcal{G}$ by B^T, and a descent step is defined by setting

$$d{=}-\lambda B^T\mathcal{G},\;\; \delta\mathcal{S}{=}Bd{=}-\lambda BB^T\mathcal{G}$$

where λ is sufficiently small and positive. The coefficients of B can be renormalized to produce unit row sums. With a uniform mesh spacing in the computational domain this formula is equivalent to the use of a gradient modified by two passes of the explicit smoothing procedure

$$\bar{\mathcal{G}}_{i,k}{=}\frac{1}{6}\mathcal{G}_{i-1,k} + \frac{2}{3}\mathcal{G}_{i,k} + \frac{1}{6}\mathcal{G}_{i+1,k}$$

with a similar smoothing procedure in the k discretization.

Implicit smoothing may also be used. The smoothing equation

$$-\epsilon_{i+\frac{1}{2},k}\left(\bar{\mathcal{G}}_{i+1,k} - \bar{\mathcal{G}}_{i,k}\right) + \epsilon_{i-\frac{1}{2},k}\left(\bar{\mathcal{G}}_{i,k} - \bar{\mathcal{G}}_{i-1,k}\right){=}\mathcal{G}_{i,k}$$

approximates the differential equation

$$\bar{\mathcal{G}} - \frac{\partial}{\partial\xi}\epsilon\frac{\partial\bar{\mathcal{G}}}{\partial\xi}{=}\mathcal{G}$$

If one sets $\delta\mathcal{S}$=$-\lambda\bar{\mathcal{G}}$, then to first order the change in the cost is

$$\begin{aligned} \delta I &= -\int\int\mathcal{G}\delta\mathcal{S}\,d\xi d\eta \\ &= -\frac{1}{\lambda}\int\int\left(\bar{\mathcal{G}} - \frac{\partial}{\partial\xi}\epsilon\frac{\partial\bar{\mathcal{G}}}{\partial\xi}\right)\bar{\mathcal{G}}\,d\xi d\eta \\ &= -\frac{1}{\lambda}\int\int\left(\bar{\mathcal{G}}^2 + \epsilon\left(\frac{\partial\bar{\mathcal{G}}}{\partial\xi}\right)^2\right)d\xi d\eta \\ &< 0, \end{aligned}$$

assuring an improvement if λ is sufficiently small and positive, unless the process has already reached a stationary point at which $\mathcal{G}$=0.

6.4 Design of Swept Wings for Very Low Shock Drag

The method has been used to carry out a study of swept wing designs which might be appropriate for long range transport aircraft. Since three dimensional calculations require substantial computational resources, it is extremely important for the practical implementation of the method to use fast solution algorithms for the flow and the adjoint equations. In this case the author's FLO87 computer program has been used as the basis of the design method. FLO87 solves the three dimensional Euler equations with a cell-centered finite volume scheme, and uses residual averaging and multigrid acceleration to obtain very rapid steady state solutions, usually in 25 to 50 multigrid cycles [66, 70]. Upwind biasing is used to produce non-oscillatory solutions, and assure the clean capture of shock waves. This is introduced through the addition of carefully controlled numerical diffusion terms, with a magnitude of order Δx^3 in smooth parts of the flow. The adjoint equations are treated in the same way as the flow equations. The fluxes are first estimated by central differences, and then modified by downwind biasing through numerical diffusive terms which are supplied by the same subroutines that were used for the flow equations.

The study has been focussed on wings designed for cruising at Mach .85, with lift coefficients in the range of .5 to .55. In every case, the wing planform was fixed while the sections were free to be changed arbitrarily by the design method, with a restriction on the minimum thickness. The initial wing has a unit-semi-span, with 38 degrees leading edge sweep. It has a modified trapezoidal planform, with straight taper from a root chord of 0.38, and a curved trailing edge in the inboard region blending into straight taper outboard of the 30 percent span station to a tip chord of 0.10, with an aspect ratio of 9.0. The initial wing sections were based on a section specifically designed by the author's two dimensional design method [73] to give shock free flow at Mach 0.78 with a lift coefficient of 0.6. This section, which has a thickness to chord ratio of 9.5 percent, was used at the tip. Similar sections with an increased thickness were used inboard. The variation of thickness was non-linear with a more rapid increase near the root, where the thickness to chord ratio of the basic section was multiplied by a factor of 1.47. The inboard sections were rotated upwards to give the initial wing 3.5 degrees twist from root to tip. The two-dimensional pressure distribution of the basic wing section at its design point was introduced as a target pressure distribution uniformly across the span. This target is presumably not realizable, but serves to favor the establishment of a relatively benign pressure distribution. The total inviscid drag coefficient, due to the combination of vortex and shock wave drag, was also included in the cost function. Since the main objective of the study was to minimize the drag, the target pressure distribution was reset after every fourth design cycle to a distribution derived by smoothing the existing pressure distribution. This allows the scheme more freedom to make changes which reduce drag. The calculations were performed with the lift coefficient forced to approach a fixed value by adjusting the angle of attack every fifth iteration of the flow solution. It was found that the computational costs can be reduced by using only 15 multigrid cycles in each flow solution, and in each adjoint solution. Although this is not enough for full convergence, it proves sufficient to provide a shape modification which leads to an improvement.

Figures 27 and 28 show a wing which was designed for a lift coefficient of .50 at Mach .85. In order to prevent the final wing from becoming too thin the threshold $\mathcal{S}_0\,(\xi,\eta)$ was set at three quarters of the height of the bump $\mathcal{S}\,(\xi,\eta)$ defining the initial wing. This calculation was performed on a mesh with 192 intervals in the ξ direction wrapping around the wing, 32 intervals in the normal η direction and 48 intervals in the spanwise ζ direction, giving a total of 294912 cells. The wing was specified by 33 sections, each with 128 points, giving a total of 4224 design variables. The plots show the initial wing geometry and pressure distribution, and the modified geometry and pressure distribution after 40 design cycles. The total inviscid drag coefficient was reduced from 0.0210 to 0.0112. The initial design exhibits a very strong shock wave in the inboard region. It can be seen that this is completely eliminated, leaving a very weak shock wave in the outboard region. To verify the solution, the final geometry was analyzed with another method, using the computer program FLO67. This program uses a cell-vertex formulation, and has recently been modified to incorporate a local extremum diminishing algorithm with a very low level of numerical diffusion [76]. When run to full convergence it was found that a better estimate of the drag coefficient of the redesigned wing is 0.0094 at Mach 0.85 with a lift coefficient of 0.5, giving a lift to drag ratio of 53. The results from FLO67 for the initial and final wings are illustrated in Figures 29 and 30. A calculation at Mach 0.500 shows a drag coefficient of 0.0087 for a lift coefficient of 0.5. Since in this case the flow is entirely subsonic, this provides an estimate of the vortex drag for this planform and lift distribution, which is just what one obtains from the standard formula for induced drag, $C_D{=}{C_L}^2/\epsilon\pi AR$, with an aspect ratio AR=9, and an efficiency factor ϵ=0.97. Thus the design method has reduced the shock wave drag coefficient to about 0.0007 at a lift coefficient of 0.5. Figure 31 shows the result of an analysis for an off design point with the Mach number increased to .86 with the same lift coefficient of .5. This results in a flat-topped pressure distribution terminating with a weak shock of nearly uniform strength across the whole span. The drag coefficient is .0097. The penalty of .0003 is so small that this might be a preferred cruising condition.

A second wing was designed in exactly the same manner as the first, starting from the same initial geometry and with the same constraints, to give a lift coefficient of .55 at Mach .85. This produces stronger shock waves and is therefore a more severe test of the method. In this case the total inviscid drag coefficient was reduced from 0.0243 to 0.0134 in 40 design cycles. Again the performance of the final design was verified by a calculation with FLO67, and when the result was fully converged the drag coefficient was found to be 0.0115. A subsonic calculation at Mach .500 shows a drag coefficient of 0.0107 for a lift coefficient of 0.55. Thus in this case the shock wave drag coefficient is about 0.0008. For a representative transport aircraft the parasite drag coefficient of the wing due to skin friction is about 0.0045. Also the fuselage drag coefficient is about 0.0050, the nacelle drag coefficient is about 0.0015, the empennage drag coefficient is about 0.0020, and excrescence drag coefficient is about 0.0010. This would give a total drag coefficient C_D=0.0255 for a lift coefficient of 0.55, corresponding to a lift to drag ratio L/D=21.6. This would be a substantial improvement over the values obtained by currently flying transport aircraft.

6.5 Optimization of Complex Configurations

In order to treat more complex configurations one can use a numerical grid generation procedure to produce a body-fitted mesh for the initial geometry, and then modify the mesh in subsequent design cycles by an analytic perturbation formula. In the two-dimensional case, for example, with computational coordinates ξ,η, let the boundary displacement at η=0 be $\delta x_b\,(\xi)$, $\delta y_b\,(\xi)$. Then the mesh points along the radial coordinate lines ξ=constant can be replaced by

$$\begin{aligned}\delta x\,(\xi,\eta) &= \mathcal{R}(\eta)\,\delta x_b\,(\xi)\\ \delta y\,(\xi,\eta) &= \mathcal{R}(\eta)\,\delta y_b\,(\xi)\end{aligned}$$

yielding

$$\delta K = \begin{bmatrix} \mathcal{R}(\eta)\,\frac{\partial}{\partial\xi}\delta x_b & \frac{d\mathcal{R}}{d\eta}\delta x_b \\ \mathcal{R}(\eta)\,\frac{\partial}{\partial\xi}\delta y_b & \frac{d\mathcal{R}}{d\eta}\delta y_b \end{bmatrix}$$

Such a procedure has been implemented by J. Reuther for the three-dimensional Euler equations, and applied to the optimization of wing-body configurations [143].

It is also possible to show that in the continuous limit the field integral in equation (57) can be eliminated. Let the change in the coordinates x_i at fixed ξ be $\delta x_i(\xi)$. Then, using the fact that the fluxes $f_j(w)$ satisfy the flow equation (48), it is possible to show by a direct calculation that

$$\frac{\partial}{\partial \xi_i}\delta Q_{ij} f_j = \frac{\partial}{\partial \xi_i} Q_{ij} \frac{\partial f_j}{\partial w}\frac{\partial w}{\partial \xi_k}\delta \xi_k$$

where

$$\delta\xi = K^{-1}\delta x.$$

A detailed derivation is given in reference [78]. Thus the perturbation equation can be written as

$$\frac{\partial}{\partial \xi_i}\left\{\frac{\partial f_j}{\partial w}\left(\delta w + \delta w^*\right)\right\} = 0$$

where δw is the variation in the solution at fixed ξ caused by the change in the boundary, while δw^* is the change in the original solution $w(\xi)$ corresponding to the mesh movement $\delta x(\xi)$

$$\delta w_i^* = \frac{\partial w_i}{\partial \xi_k}\delta \xi_k$$

Now

$$\begin{aligned}\int \psi^T \frac{\partial}{\partial \xi_i}\delta F_i d\xi &= \int_B n_i \psi^T \delta F_i d\xi_B \\ &- \int_D \frac{\partial \psi^T}{\partial \xi_i} C_i \left(\delta w + \delta w^*\right) d\xi\end{aligned}$$

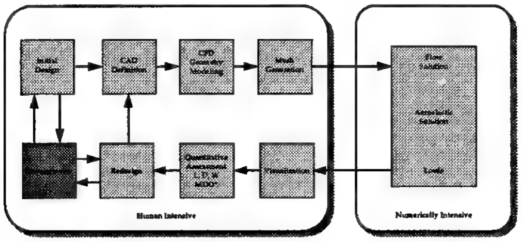
and if ψ satisfies the adjoint equation the entire field integral is eliminated, leaving only the boundary integral in equation (57).

In an actual discretization the field terms are not zero, but this result suggests that they should be small if a fine enough mesh is used, and might be dropped. This allows a drastic simplification of the treatment of complex configurations. Preliminary numerical experiments with airfoil and wing calculations indicate roughly the same convergence with and without the field terms in the gradient.

7. OUTLOOK AND CONCLUSIONS

Better algorithms and better computer hardware have contributed about equally to the progress of computational science in the last two decades. In 1970 the Control Data 6600 represented the state of the art in computer hardware with a speed of about 10^6 operations per second (one megaflop), while in 1990 the 8 processor Cray YMP offered a performance of about 10^9 operations per second (one gigaflop). Correspondingly, steady-state Euler calculations which required 5,000–10,000 steps prior to 1980 could be performed in 10–50 steps in 1990 using multigrid acceleration. With the advent of massively parallel computers it appears that the progress of computer hardware may even accelerate. Teraflop machines offering further improvement by a factor of 1,000 are likely to be available within a few years. Parallel architectures will force a reappraisal of existing algorithms, and their effective utilization will require the extensive development of new parallel software.

In parallel with the transition to more sophisticated algorithms, the present challenge is to extend the effective use of CFD to more complex applications. A key problem is the treatment of multiple space and time scales. These arise not only in turbulent flows, but also in many other situations such as chemically reacting flows, combustion, flame fronts and plasma dynamics. Another challenge, is presented by problems with moving boundaries. Examples include helicopter rotors, and rotor-stator interaction in turbomachinery. Algorithms for these problems can be significantly improved by innovative concepts, such as the idea of time inclining. It can be anticipated that interdisciplinary applications in which CFD is coupled with the computational analysis of other properties of the design will play an increasingly important role. These applications may include structural, thermal and electromagnetic analysis. Aeroelastic problems and integrated control system and aerodynamic design are likely target areas. The development of improved algorithms continues to be important in providing the basic building blocks for numerical simulation. In particular, better error estimation procedures must be developed and incorporated in the simulation software to provide error control. The basic simulation software is only one of the needed ingredients, however. The flow solver must be embedded in a user-friendly system for geometry modeling, output analysis, and data management that will provide a complete numerical design environment. These are the ingredients which are needed for the full realization of the concept of a numerical wind tunnel. Figures 19 and 20 illustrate the way in which a numerical wind tunnel might evolve from current techniques, which involve massive data handling tasks, to a fully integrated design environment.



*MDO: Multi-Disciplinary Optimization

Figure 19: Concept for a numerical wind tunnel.

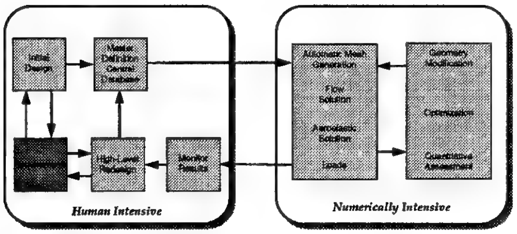


Figure 20: Advanced numerical wind tunnel.

In the long run, computational simulation should become the principal tool of the aerodynamic design process because of the flexibility it provides for the rapid and comparatively inexpensive evaluation of alternative designs, and because it can be integrated in a numerical design environment providing for both multi-disciplinary analysis and multi-disciplinary optimization.

REFERENCES

[1] R. Abid, C.G. Speziale, and S. Thangam. Application of a new k-τ model to near wall turbulent flows. AIAA Paper 91-0614, AIAA 29th Aerospace Sciences Meeting, Reno, NV, January 1991.

[2] H. Aiso. Admissibility of difference approximations for scalar conservation laws. *Hiroshima Math. Journal*, 23:15–61, 1993.

[3] J. J. Alonso and A. Jameson. Fully-implicit time-marching aeroelastic solutions. *AIAA paper 94-0056*, AIAA 32nd Aerospace Sciences Meeting, Reno, Nevada, January 1994.

[4] J. J. Alonso, L. Martinelli, and A. Jameson. Multigrid unsteady Navier-Stokes calculations with aeroelastic applications. *AIAA paper 95-0048*, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995.

[5] B.K. Anderson, J.L. Thomas, and B. Van Leer. A comparison of flux vector splittings for the Euler equations. AIAA Paper 85-0122, Reno, NV, January 1985.

[6] W.K. Anderson, J.L. Thomas, and D.L. Whitfield. Multigrid acceleration of the flux split Euler equations. AIAA Paper 86-0274, AIAA 24th Aerospace Sciences Meeting, Reno, January 1986.

[7] T.J. Baker. Mesh generation by a sequence of transformations. *Appl. Num. Math.*, 2:515–528, 1986.

[8] N. Balakrishnan and S. M. Deshpande. New upwind schemes with wave-particle splitting for inviscid compressible flows. *Report 91 FM 12*, Indian Institute of Science, 1991.

[9] B. Baldwin and H. Lomax. Thin layer approximation and algebraic model for separated turbulent flow. AIAA Paper 78-257, 1978.

[10] B.S. Baldwin and T.J. Barth. A one-equation turbulence transport model for high Reynolds number wall-bounded flows. AIAA Paper 91-0610, AIAA 29th Aerospace Sciences Meeting, Reno, NV, January 1991.

[11] T. J. Barth. Aspects of unstructured grids and finite volume solvers for the Euler and Navier Stokes equations. In *von Karman Institute for Fluid Dynamics Lecture Series Notes 1994-05*, Brussels, 1994.

[12] T.J. Barth and P.O. Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA paper 90-0013, January 1990.

[13] T.J. Barth and D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA paper 89-0366*, AIAA 27th Aerospace Sciences Meeting, Reno, Nevada, January 1989.

[14] J.T. Batina. Implicit flux-split Euler schemes for unsteady aerodynamic analysis involving unstructured dynamic meshes. AIAA paper 90-0936, April 1990.

[15] F. Bauer, P. Garabedian, D. Korn, and A. Jameson. *Supercritical Wing Sections II.* Springer Verlag, New York, 1975.

[16] O. Baysal and M. E. Eleshaky. Aerodynamic design optimization using sensitivity anaysis and computational fluid dynamics. *AIAA paper 91-0471*, 29th Aerospace Sciences Meeting, Reno, Nevada, January 1991.

[17] R.W. Beam and R.F. Warming. An implicit finite difference algorithm for hyperbolic systems in conservation form. *J. Comp. Phys.*, 23:87–110, 1976.

[18] A. Belov, L. Martinelli, and A. Jameson. A new implicit algorithm with multigrid for unsteady incompressible flow calculations. *AIAA paper 95-0049*, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995.

[19] J.A. Benek, P.G. Buning, and J.L. Steger. A 3-D Chimera grid embedding technique. In *Proceedings AIAA 7th Computational Fluid Dynamics Conference*, pages 507–512, Cincinnati, OH, 1985. AIAA Paper 85-1523.

[20] J.A. Benek, T.L. Donegan, and N.E. Suhs. Extended Chimera grid embedding scheme with applications to viscous flows. AIAA Paper 87-1126, AIAA 8th Computational Fluid Dynamics Conference, Honolulu, HI, 1987.

[21] M. Berger and A. Jameson. Automatic adaptive grid refinement for the Euler equations. *AIAA Journal*, 23:561–568, 1985.

[22] M. Berger and R.J. LeVeque. An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries. AIAA Paper 89-1930, 1989.

[23] J.P. Boris and D.L. Book. Flux corrected transport, 1 SHASTA, a fluid transport algorithm that works. *J. Comp. Phys.*, 11:38–69, 1973.

[24] A. Brandt. Multi-level adaptive solutions to boundary value problems. *Math. Comp.*, 31:333–390, 1977.

[25] M.O. Bristeau, R. Glowinski, J. Periaux, P. Perrier, O. Pironneau, and C. Poirier. On the numerical solution of nonlinear problems in fluid dynamics by least squares and finite element methods (II), application to transonic flow simulations. *Comp. Meth. Appl. Mech. and Eng.*, 51:363–394, 1985.

[26] R.J. Busch, Jr. Computational fluid dynamics in the design of the Northrop/McDonnell Douglas YF-23 ATF prototype. *AIAA paper 91-1627*, AIAA 21st Fluid Dynamics, Plasmadynamics & Lasers Conference, Honolulu, Hawaii, 1991.

[27] C. Canuto, M.Y. Hussaini, A. Quarteroni, and D.A. Zang. *Spectral Methods in Fluid Dynamics.* Springer Verlag, 1987.

[28] M.H. Carpenter, D. Gottlieb, and S. Abarbanel. Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes. ICASE Report 93-9, Hampton, VA, March 1993.

[29] D.A. Caughey. A diagonal implicit multigrid algorithm for the Euler equations. AIAA Paper 87-453, 25th Aerospace Sciences Meeting, Reno, January 1987.

[30] T. Cebeci and A.M.O. Smith. *Analysis of Turbulent Boundary Layers.* Academic Press, 1974.

[31] S.R. Chakravarthy. Relaxation methods for unfactored implicit upwind schemes. AIAA Paper 84-0165, AIAA 22nd Aerospace Sciences Meeting, Reno, January 1984.

[32] S.R. Chakravarthy, A. Harten, and S. Osher. Essentially non-oscillatory shock capturing schemes of uniformly very high accuracy. AIAA Paper 86-0339, AIAA 24th Aerospace Sciences Meeting, Reno, January 1986.

[33] R. Chipman and A. Jameson. Fully conservative numerical solutions for unsteady irrotational transonic flow about airfoils. AIAA Paper 79-1555, AIAA 12th Fluid and Plasma Dynamics Conference, Williamsburg, VA, July 1979.

[34] S.E. Cliff and S.D. Thomas. Euler/experimental correlations of sonic boom pressure signatures. AIAA Paper 91-3276, AIAA 9th Applied Aerodynamics Conference, Baltimore, September 1991.

[35] T.J. Coakley. Numerical simulation of viscous transonic airfoil flows. AIAA Paper 87-0416, AIAA 25th Aerospace Sciences Meeting, Reno, January 1987.

[36] J.P. Croisille and P. Villedieu. Kinetic flux splitting schemes for hypersonic flows. In M. Napolitano and F. Sobetta, editors, *Proc 13th International Congress on Numerical Methods in Fluid Dynamics*, pages 310–313, Rome, July 1992. Springer Verlag.

[37] R.M. Cummings, Y.M. Rizk, L.B. Schiff, and N.M. Chaderjian. Navier-Stokes predictions for the F-18 wing and fuselage at large incidence. *J. of Aircraft*, 29:565–574, 1992.

[38] G. Dahlquist. A special stability problem for linear multistep methods. *BIT*, 3:27–43, 1963.

[39] J.F. Dannenhoffer and J.R. Baron. Robust grid adaptation for complex transonic flows. AIAA Paper 86-0495, AIAA 24th Aerospace Sciences Meeting, Reno, January 1986.

[40] D. Deganiand and L. Schiff. Computation of turbulent supersonic flows around pointed bodies having crossflow separation. *J. Comp. Phys.*, 66:173–196, 1986.

[41] B. Delaunay. Sur la sphère vide. *Bull. Acad. Science USSR VII: Class Scil, Mat. Nat.*, pages 793–800, 1934.

[42] S.M. Deshpande. On the Maxwellian distribution, symmetric form and entropy conservation for the Euler equations. NASA TP 2583, 1986.

[43] A. Eberle. A finite volume method for calculating transonic potential flow around wings from the minimum pressure integral. NASA TM 75324, 1978. Translated from MBB UFE 1407(0).

[44] P.R. Eiseman. A multi-surface method of coordinate generation. *J. Comp. Phys.*, 33:118–150, 1979.

[45] W. Eppard and B. Grossman. A multi-dimensional kinetic-based upwind solver for the Euler equations. AIAA Paper 93-3303, AIAA 11th Computational Fluid Dynamics Conference, Orlando, FL, July 1993.

[46] L.E. Eriksson. Generation of boundary-conforming grids around wing-body configurations using transfinite interpolation. *AIAA Journal*, 20:1313–1320, 1982.

[47] J. Farmer, L. Martinelli, A. Jameson, and G. Cowles. Fully-nonlinear CFD techniques for ship performance analysis and design. *AIAA paper 95-1690*, AIAA 12th Computational Fluid Dynamics Conference, San Diego, CA, June 1995.

[48] R.P. Fedorenko. The speed of convergence of one iterative process. *USSR Comp. Math. and Math. Phys.*, 4:227–235, 1964.

[49] C.W. Gear. The numerical integration of stiff ordinary differential equations. Report 221, University of Illinois Department of Computer Science, 1967.

[50] M. Giles, M. Drela, and W.T. Thompkins. Newton solution of direct and inverse transonic Euler equations. AIAA Paper 85-1530, Cincinnati, 1985.

[51] S.K. Godunov. A difference method for the numerical calculation of discontinuous solutions of hydrodynamic equations. *Mat. Sbornik*, 47:271–306, 1959. Translated as JPRS 7225 by U.S. Dept. of Commerce, 1960.

[52] M.H. Ha. The impact of turbulence modelling on the numerical prediction of flows. In M. Napolitano and F. Solbetta, editors, *Proc. of the 13th International Conference on Numerical Methods in Fluid Dynamics*, pages 27–46, Rome, Italy, July 1992. Springer Verlag, 1993.

[53] W. Hackbusch. On the multi-grid method applied to difference equations. *Computing*, 20:291–306, 1978.

[54] M. Hafez, J.C. South, and E.M. Murman. Artificial compressibility method for numerical solutions of the transonic full potential equation. *AIAA Journal*, 17:838–844, 1979.

[55] M.G. Hall. Cell vertex multigrid schemes for solution of the Euler equations. In *Proc. IMA Conference on Numerical Methods for Fluid Dynamics*, Reading, April 1985.

[56] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comp. Phys.*, 49:357–393, 1983.

[57] P.W. Hemker and S.P. Spekreijse. Multigrid solution of the steady Euler equations. In *Proc. Oberwolfach Meeting on Multigrid Methods*, December 1984.

[58] J.L. Hess and A.M.O. Smith. Calculation of non-lifting potential flow about arbitrary three-dimensional bodies. Douglas Aircraft Report ES 40622, 1962.

[59] C. Hirsch, C. Lacol, and H. Deconinck. Convection algorithms based on a diagonalization procedure for the multi-dimensional Euler equations. In *Proc. AIAA 8th Computational Fluid Dynamics Conference*, pages 667–676, Hawaii, June 1987. AIAA Paper 87-1163.

[60] C. Hirsch and P. Van Ransbeeck. Multi-dimensional upwinding and artificial dissipation. Technical report. Published in *Frontiers of Computational Fluid Dynamics 1994*, D. A. Caughey and M. M. Hafez, editors, Wiley, pp. 597–626.

[61] D.G. Holmes and S.H. Lamson. Adaptive triangular meshes for compressible flow solutions. In *Proceedings First International Conference on Numerical Grid Generation in Computational Fluid Dynamics*, pages 413–424, Landshut, FRG, July 1986.

[62] T.J.R. Hughes, L.P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics,I, Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Comp. Meth. Appl. Mech. and Eng.*, 59:223–231, 1986.

[63] A. Jameson. Iterative solution of transonic flows over airfoils and wings, including flows at Mach 1. *Comm. on Pure and Appl. Math.*, 27:283–309, 1974.

[64] A. Jameson. Transonic potential flow calculations in conservation form. In *Proc. AIAA 2nd Computational Fluid Dynamics Conference*, pages 148–161, Hartford, 1975.

[65] A. Jameson. Solution of the Euler equations by a multigrid method. *Appl. Math. and Comp.*, 13:327–356, 1983.

[66] A. Jameson. Solution of the Euler equations for two dimensional transonic flow by a multigrid method. *Appl. Math. Comp.*, 13:327–356, 1983.

[67] A. Jameson. Multigrid algorithms for compressible flow calculations. In *Second European Conference on Multigrid Methods*, Cologne, October 1985. Princeton University Report MAE 1743.

[68] A. Jameson. Non-oscillatory shock capturing scheme using flux limited dissipation. In B.E. Engquist, S. Osher, and R.C.J. Sommerville, editors, *Lectures in Applied Mathematics, Vol. 22, Part 1, Large Scale Computations in Fluid Mechanics*, pages 345–370. AMS, 1985.

[69] A. Jameson. Transonic flow calculations for aircraft. In F. Brezzi, editor, *Lecture Notes in Mathematics, Numerical Methods in Fluid Dynamics*, pages 156–242. Springer Verlag, 1985.

[70] A. Jameson. Multigrid algorithms for compressible flow calculations. In W. Hackbusch and U. Trottenberg, editors, *Lecture Notes in Mathematics, Vol. 1228*, pages 166–201. Proceedings of the 2nd European Conference on Multigrid Methods, Cologne, 1985, Springer-Verlag, 1986.

[71] A. Jameson. A vertex based multigrid algorithm for three-dimensional compressible flow calculations. In T.E. Tezduar and T.J.R. Hughes, editors, *Numerical Methods for Compressible Flow - Finite Difference, Element And Volume Techniques*, 1986. ASME Publication AMD 78.

[72] A. Jameson. Aerodynamic design via control theory. *J. Sci. Comp.*, 3:233–260, 1988.

[73] A. Jameson. Automatic design of transonic airfoils to reduce the shock induced pressure drag. In *Proceedings of the 31st Israel Annual Conference on Aviation and Aeronautics, Tel Aviv*, pages 5–17, February 1990.

[74] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA paper 91-1596*, AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 1991.

[75] A. Jameson. Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence in transonic and hypersonic flows. AIAA Paper 93-3359, AIAA 11th Computational Fluid Dynamics Conference, Orlando, FL, July 1993.

[76] A. Jameson. Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence in transonic and hypersonic flow. *AIAA paper 93-3359*, AIAA 11th Computational Fluid Dynamics Conference, Orlando, Florida, July 1993.

[77] A. Jameson. Optimum aerodynamic design via boundary control. Technical report, AGARD FDP/Von Karman Institute Special Course on Optimum Design Methods in Aerodynamics, Brussels, April 1994.

[78] A. Jameson. MAE Technical Report 2050, Princeton University, Princeton, New Jersey, October 1995.

[79] A. Jameson. Analysis and design of numerical schemes for gas dynamics 1, artificial diffusion, upwind biasing, limiters and their effect on multigrid convergence. *Int. J. of Comp. Fluid Dyn.*, 4:171–218, 1995.

[80] A. Jameson. Analysis and design of numerical schemes for gas dynamics 2, artificial diffusion and discrete shock structure. *Int. J. of Comp. Fluid Dyn.*, To Appear.

[81] A. Jameson and T.J. Baker. Improvements to the aircraft Euler method. AIAA Paper 87-0452, AIAA 25th Aerospace Sciences Meeting, Reno, January 1987.

[82] A. Jameson, T.J. Baker, and N.P. Weatherill. Calculation of inviscid transonic flow over a complete aircraft. AIAA Paper 86-0103, AIAA 24th Aerospace Sciences Meeting, Reno, January 1986.

[83] A. Jameson and D.J. Mavriplis. Multigrid solution of the Euler equations on unstructured and adaptive grids. In S. McCormick, editor, *Multigrid Methods, Theory, Applications and Supercomputing. Lecture Notes in Pure and Applied Mathematics*, volume 110, pages 413–430, April 1987.

[84] A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. AIAA Paper 81-1259, 1981.

[85] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods with Runge-Kutta time stepping schemes. *AIAA paper 81-1259*, January 1981.

[86] A. Jameson and E. Turkel. Implicit schemes and LU decompositions. *Math. Comp.*, 37:385–397, 1981.

[87] M. Jayaram and A. Jameson. Multigrid solution of the Navier-Stokes equations for flow over wings. *AIAA paper 88-0705*, AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, January 1988.

[88] D. Johnson and L. King. A mathematically simple turbulence closure model for attached and separated turbulent boundary layers. *AIAA Journal*, 23:1684–1692, 1985.

[89] W.P. Jones and B.E. Launder. The calculation of low-Reynolds-number phenomena with a two-equation model of turbulence. *Int. J. of Heat Tran.*, 16:1119–1130, 1973.

[90] W.H. Jou. Boeing Memorandum AERO-B113B-L92-018, September 1992. To Joseph Shang.

[91] T.J. Kao, T.Y. Su, and N.J. Yu. Navier-Stokes calculations for transport wing-body configurations with nacelles and struts. AIAA Paper 93-2945, AIAA 24th Fluid Dynamics Conference, Orlando, July 1993.

[92] G.E. Karniadakis and S.A. Orszag. Nodes, modes and flow codes. *Physics Today*, pages 34–42, March 1993.

[93] M.H. Lallemand and A. Dervieux. A multigrid finite-element method for solving the two-dimensional Euler equations. In S.F. McCormick, editor, *Proceedings of the Third Copper Mountain Conference on Multigrid Methods, Lecture Notes in Pure and Applied Mathematics*, pages 337–363, Copper Mountain, April 1987.

[94] A.M. Landsberg, J.P. Boris, W. Sandberg, and T.R. Young. Naval ship superstructure design: Complex three-dimensional flows using an efficient, parallel method. *High Performance Computing 1993: Grand Challenges in Computer Simulation*, 1993.

[95] P. D. Lax. Hyperbolic systems of conservation laws. *SIAM Regional Series on Appl. Math.*, II, 1973.

[96] P.D. Lax. Hyperbolic systems of conservation laws and the mathematical theory of shock waves. *SIAM Regional Series on Appl. Math.*, 11, 1973.

[97] P.D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure. Appl. Math.*, 13:217–237,1960.

[98] B. Van Leer. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second order scheme. *J. Comp. Phys.*, 14:361–370, 1974.

[99] B. Van Leer. Towards the ultimate conservative difference scheme. III upstream-centered finite-difference schemes for ideal compressible flow. *J. Comp. Phys.*, 23:263–275, 1975.

[100] B. Van Leer. Flux vector splitting for the Euler equations. In E. Krause, editor, *Proceedings of the 8th International Conference on Numerical Methods in Fluid Dynamics*, pages 507–512, Aachen, 1982.

[101] B. Van Leer. Progress in multi-dimensional upwind differencing. In M. Napolitano and F. Solbetta, editors, *Proc. 13th International Conference on Numerical Methods in Fluid Dynamics*, pages 1–26, Rome, July 1992. Springer Verlag, 1993.

[102] B. Van Leer, W. T. Lee, and P. L. Roe. Characteristic time stepping or local preconditioning of the Euler equations. *AIAA paper 91-1552*, AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 1991.

[103] D. Lefebre, J.Peraire, and K.Morgan. Finite element least squares solutions of the Euler equations using linear and quadratic approximations. *Int. J. Comp. Fluid Dynamics*, 1:1–23, 1993.

[104] S.K. Lele. Compact finite difference schemes with spectral-like resolution. CTR Manuscript 107, 1990.

[105] J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, 1971. Translated by S.K. Mitter.

[106] M-S. Liou and C.J. Steffen. A new flux splitting scheme. *J. Comp. Phys.*, 107:23–39, 1993.

[107] F. Liu and A. Jameson. Multigrid Navier-Stokes calculations for three-dimensional cascades. *AIAA paper 92-0190*, AIAA 30th Aerospace Sciences Meeting, Reno, Nevada, January 1992.

[108] R. Löhner and D. Martin. An implicit linelet-based solver for incompressible flows. AIAA paper 92-0668, AIAA 30th Aerospace Sciences Meeting, Reno, NV, January 1992.

[109] R. Löhner, K. Morgan, and J. Peraire. Improved adaptive refinement strategies for the finite element aerodynamic configurations. AIAA Paper 86-0499, AIAA 24th Aerospace Sciences Meeting, Reno, January 1986.

[110] R. Löhner, K. Morgan, J. Peraire, and O.C. Zienkiewicz. Finite element methods for high speed flows. In *Proc. AIAA 7th Computational Fluid Dynamics Conference*, Cincinnati, OH, 1985. AIAA Paper 85-1531.

[111] R. Löhner and P. Parikh. Generation of three-dimensional unstructured grids by the advancing front method. AIAA Paper 88-0515, Reno, NV, January 1988.

[112] R.W. MacCormack and A.J. Paullay. Computational efficiency achieved by time splitting of finite difference operators. AIAA Paper 72-154, 1972.

[113] A. Majda and S. Osher. Numerical viscosity and the entropy condition. *Comm. on Pure Appl. Math.*, 32:797–838, 1979.

[114] L. Martinelli and A. Jameson. Validation of a multigrid method for the Reynolds averaged equations. *AIAA paper 88-0414*, 1988.

[115] L. Martinelli, A. Jameson, and E. Malfa. Numerical simulation of three-dimensional vortex flows over delta wing configurations. In M. Napolitano and F. Solbetta, editors, *Proc. 13th International Confrence on Numerical Methods in Fluid Dynamics*, pages 534–538, Rome, Italy, July 1992. Springer Verlag, 1993.

[116] L. Martinelli and V. Yakhot. RNG-based turbulence transport approximations with applications to transonic flows. AIAA Paper 89-1950, AIAA 9th Computational Fluid Dynamics Conference, Buffalo, NY, June 1989.

[117] D.J. Mavriplis and A. Jameson. Multigrid solution of the Navier-Stokes equations on triangular meshes. *AIAA Journal*, 28(8):1415–1425, August 1990.

[118] D.J. Mavriplis and L. Martinelli. Multigrid solution of compressible turbulent flow on unstructured meshes using a two-equation model. AIAA Paper 91-0237, January 1991.

[119] N. D. Melson, M. D. Sanetrik, and H. L. Atkins. Time-accurate Navier-Stokes calculations with multigrid acceleration. In *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods*, Copper Mountain, April 1993.

[120] J.E. Melton, S.A. Pandya, and J.L. Steger. 3D Euler flow solutions using unstructured Cartesian and prismatic grids. AIAA Paper 93-0331, Reno, NV, January 1993.

[121] F. Menter. Zonal two-equation k-ω turbulence models for aerodynamic flows. AIAA Paper 93-2906, AIAA 24th Fluid Dynamics Meeting, Orlando, July 1993.

[122] E.M. Murman. Analysis of embedded shock waves calculated by relaxation methods. *AIAA Journal*, 12:626–633, 1974.

[123] E.M. Murman and J.D. Cole. Calculation of plane steady transonic flows. *AIAA Journal*, 9:114–121, 1971.

[124] R.H. Ni. A multiple grid scheme for solving the Euler equations. *AIAA Journal*, 20:1565–1571, 1982.

[125] S. Obayashi and K. Kuwakara. LU factorization of an implicit scheme for the compressible Navier-Stokes equations. AIAA Paper 84-1670, AIAA 17th Fluid Dynamics and Plasma Dynamics Conference, Snowmass, June 1984.

[126] J.T. Oden, L. Demkowicz, T. Liszka, and W. Rachowicz. *h-p* adaptive finite element methods for compressible and incompressible flows. In S. L. Venneri A. K. Noor, editor, *Proceedings of the Symposium on Computational Technology on Flight Vehicles*, pages 523–534, Washington, D.C., November 1990. Pergamon.

[127] S. Orszag and D. Gottlieb. Numerical analysis of spectral methods. *SIAM Regional Series on Appl. Math.*, 26, 1977.

[128] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Num. Anal.*, 121:217–235, 1984.

[129] S. Osher and S. Chakravarthy. High resolution schemes and the entropy condition. *SIAM J. Num Anal.*, 21:955–984, 1984.

[130] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic systems of conservation laws. *Math. Comp.*, 38:339–374, 1982.

[131] S. Osher and E. Tadmor. On the convergence of difference approximations to scalar conservation laws. *Math. Comp.*, 50:19–51, 1988.

[132] H. Paillère and H. Deconinck. A review of multidimensional upwind residual distribution schemes for the euler equations. *To appear in CFD Review*, 1995.

[133] N. A. Pierce and M. B. Giles. Preconditioning on stretched meshes. Report 95/10 1995, Oxford University Computing Laboratory, Oxford, December 1995.

[134] O. Pironneau. *Optimal Shape Design for Elliptic Systems*. Springer-Verlag, New York, 1984.

[135] K.G. Powell and B. van Leer. A genuinely multidimensional upwind cell-vertex scheme for the Euler equations. AIAA Paper 89-0095, AIAA 27th Aerospace Sciences Meeting, Reno, January 1989.

[136] K. Prendergast and K. Xu. Numerical hydrodynamics from gas kinetic theory. *J. Comp. Phys.*, 109:53–66, November 1993.

[137] T.H. Pulliam and J.L. Steger. Implicit finite difference simulations of three-dimensional compressible flow. *AIAA Journal*, 18:159–167, 1980.

[138] J.J. Quirk. An alternative to unstructured grids for computing gas dynamics flows about arbitrarily complex two-dimensional bodies. ICASE Report 92-7, Hampton, VA, February 1992.

[139] R. Radespiel, C. Rossow, and R.C. Swanson. An efficient cell-vertex multigrid scheme for the three-dimensional Navier-Stokes equations. In *Proc. AIAA 9th Computational Fluid Dynamics Conference*, pages 249–260, Buffalo, NY, June 1989. AIAA Paper 89-1953-CP.

[140] M.M. Rai and P. Moin. Direct numerical simulation of transition and turbulence in a spatially evolving boundary layer. AIAA Paper 91-1607 CP, AIAA 10th Computational Fluid Dynamics Conference, Honolulu, HI, June 1991.

[141] S. V. Rao and S. M. Deshpande. A class of efficient kinetic upwind methods for compressible flows. *Report 91 FM 11*, Indian Institute of Science, 1991.

[142] J. Reuther and A. Jameson. Aerodynamic shape optimization of wing and wing-body configurations using control theory. *AIAA paper 95-0213*, 33rd Aerospace Sciences Meeting and Exibit, Reno, Nevada, January 1995.

[143] J. Reuther and A. Jameson. Aerodynamic shape optimization of wing and wing-body configurations using control theory. *AIAA paper 95-0123*, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995.

[144] H. Rieger and A. Jameson. Solution of steady three-dimensional compressible Euler and Navier-Stokes equations by and implicit LU scheme. *AIAA paper 88-0619*, AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, January 1988.

[145] A. Rizzi and L.E. Eriksson. Computation of flow around wings based on the Euler equations. *J. Fluid Mech.*, 148:45–71, 1984.

[146] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comp. Phys.*, 43:357–372, 1981.

[147] P.E. Rubbert and G.R. Saaris. A general three-dimensional potential flow method applied to V/STOL aerodynamics. SAE Paper 680304, 1968.

[148] C.L. Rumsey and V.N. Vatsa. A comparison of the predictive capabilities of several turbulence models using upwind and centered - difference computer codes. AIAA Paper 93-0192, AIAA 31st Aerospace Sciences Meeting, Reno, January 1993.

[149] S.S. Samant, J.E. Bussoletti, F.T. Johnson, R.H. Burkhart, B.L. Everson, R.G. Melvin, D.P. Young, L.L. Erickson, and M.D. Madson. TRANAIR: A computer code for transonic analyses of arbitrary configurations. AIAA Paper 87-0034, 1987.

[150] K. Sawada and S. Takanashi. A numerical investigation on wing/nacelle interferences of USB configuration. In *Proceedings AIAA 25th Aerospace Sciences Meeting*, Reno, NV, 1987. AIAA paper 87-0455.

[151] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comp. Phys.*, 77:439–471, 1988.

[152] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes II. *J. Comp. Phys.*, 83:32–78, 1989.

[153] C.W. Shu, T.A. Zang, G. Erlebacher, D. Whitaker, and S. Osher. High-order ENO schemes applied to two- and three-dimensional compressible flow. *Appl. Num. Math.*, 9:45–71, 1992.

[154] B. R. Smith. A near wall model for the $k - l$ two equation turbulence model. *AIAA paper 94-2386*, 25th AIAA Fluid Dynamics Conference, Colorado Springs, CO, June 1994.

[155] L.M. Smith and W.C. Reynolds. On the Yakhot-Orszag renormalization group for deriving turbulence statistics and models. *Phys. Fluids A*, 4:364–390, 1992.

[156] R.E. Smith. Three-dimensional algebraic mesh generation. In *Proc. AIAA 6th Computational Fluid Dynamics Conference*, Danvers, MA, 1983. AIAA Paper 83-1904.

[157] R.L. Sorenson. Elliptic generation of compressible three-dimensional grids about realistic aircraft. In J. Hauser and C. Taylor, editors, *International Conference on Numerical Grid Generation in Computational Fluid Dynamics*, Landshut, F. R. G., 1986.

[158] R.L. Sorenson. Three-dimensional elliptic grid generation for an F-16. In J.L. Steger and J.F. Thompson, editors, *Three-Dimensional Grid Generation for Complex Configurations: Recent Progress*, 1988. AGARDograph.

[159] P. Spalart and S. Allmaras. A one-equation turbulent model for aerodynamic flows. AIAA Paper 92-0439, AIAA 30th Aerospace Sciences Meeting, Reno, NV, January 1992.

[160] C.G. Speziale, E.C. Anderson, and R. Abid. A critical evaluation of two-equation models for near wall turbulence. AIAA Paper 90-1481, June 1990. ICASE Report 90-46.

[161] J.L. Steger and D.S. Chaussee. Generation of body-fitted coordinates using hyperbolic partial differential equations. *SIAM J. Sci. and Stat. Comp.*, 1:431–437, 1980.

[162] J.L. Steger and R.F. Warming. Flux vector splitting of the inviscid gas dynamic equations with applications to finite difference methods. *J. Comp. Phys.*, 40:263–293, 1981.

[163] B. Stoufflet, J. Periaux, F. Fezoui, and A. Dervieux. Numerical simulation of 3-D hypersonic Euler flows around space vehicles using adapted finite elements. AIAA paper 87-0560, January 1987.

[164] G. Strang and G. Fix. *Analysis of the Finite Element Method.* Prentice Hall, 1993.

[165] R.C. Swanson and E. Turkel. On central-difference and upwind schemes. *J. Comp. Phys.*, 101:297–306, 1992.

[166] P.K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Num. Anal.*, 21:995–1011, 1984.

[167] S. Ta'asan, G. Kuruvila, and M. D. Salas. Aerodynamic design and optimization in one shot. *AIAA paper 92-005*, 30th Aerospace Sciences Meeting and Exibit, Reno, Nevada, January 1992.

[168] Shlomo Ta'asan. Canonical forms of multidimensional, steady inviscid flows. ICASE 93-34, Institute for Computer Applications in Science and Engineering, Hampton, VA, 1993.

[169] E. Tadmor. Numerical viscosity and the entropy condition for conservative difference schemes. *Math. Comp.*, 32:369–382, 1984.

[170] J.F. Thompson, F.C. Thames, and C.W. Mastin. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *J. Comp. Phys.*, 15:299–319, 1974.

[171] J.F. Thompson, Z.U.A. Warsi, and C.W. Mastin. Boundary-fitted coordinate systems for numerical solution of partial differential equations: A review. *J. Comp. Phys.*, 47:1–108, 1982.

[172] E. Turkel. Preconditioned methods for solving the incompressible and low speed equations. *J. Comp. Phys.*, 72:277–298, 1987.

[173] V. Venkatakrishnan. Newton solution of inviscid and viscous problems. AIAA paper 88-0413, January 1988.

[174] V. Venkatakrishnan. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *AIAA paper 93-0880*, AIAA 31st Aerospace Sciences Meeting, Reno, Nevada, January 1993.

[175] G. Voronoï. Nouvelles applications des parametres continus à la théorie des formes quadratiques. Deuxième mémoire: Recherches sur les parallelloedres primitifs. *J. Reine Angew. Math.*, 134:198–287, 1908.

[176] Y. Wada and M-S. Liou. A flux splitting scheme with high-resolution and robustness for discontinuities. *AIAA paper 94-0083*, AIAA 32nd Aerospace Sciences Meeting, Reno, Nevada, January 1994.

[177] N.P. Weatherill and C.A. Forsey. Grid generation and flow calculations for aircraft geometries. *J. Aircraft*, 22:855–860, 1985.

[178] D.C. Wilcox. A half a century historical review of the k-ω model. AIAA Paper 91-0615, AIAA 29th Aerospace Sciences Meeting, Reno, NV, January 1991.

[179] F. Woodward. An improved method for the aerodynamic analysis of wing-body-tail configurations in subsonic and supersonic flow, Part 1 — theory and application. NASA CR 2228 Pt. 1, May 1973.

[180] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comp. Phys.*, 54:115–173, 1984.

[181] K. Xu, L. Martinelli, and A. Jameson. Gas-kinetic finite volume methods, flux-vector splitting and artificial diffusion. *J. Comp. Phys.*, 120:48–65, 1995.

[182] V. Yakhot and S.A. Orszag. Renormalization group analysis of turbulence. I. Basic theory. *J. Sci. Comp.*, 1:3–51, 1986.

[183] H.C. Yee. On symmetric and upwind TVD schemes. In *Proc. 6th GAMM Conference on Numerical Methods in Fluid Mechanics*, Gottingen, September 1985.

[184] S. Yoon and A. Jameson. Lower-upper Symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations. AIAA Paper 87-0600, AIAA 25th Aerospace Sciences Meeting, Reno, January 1987.

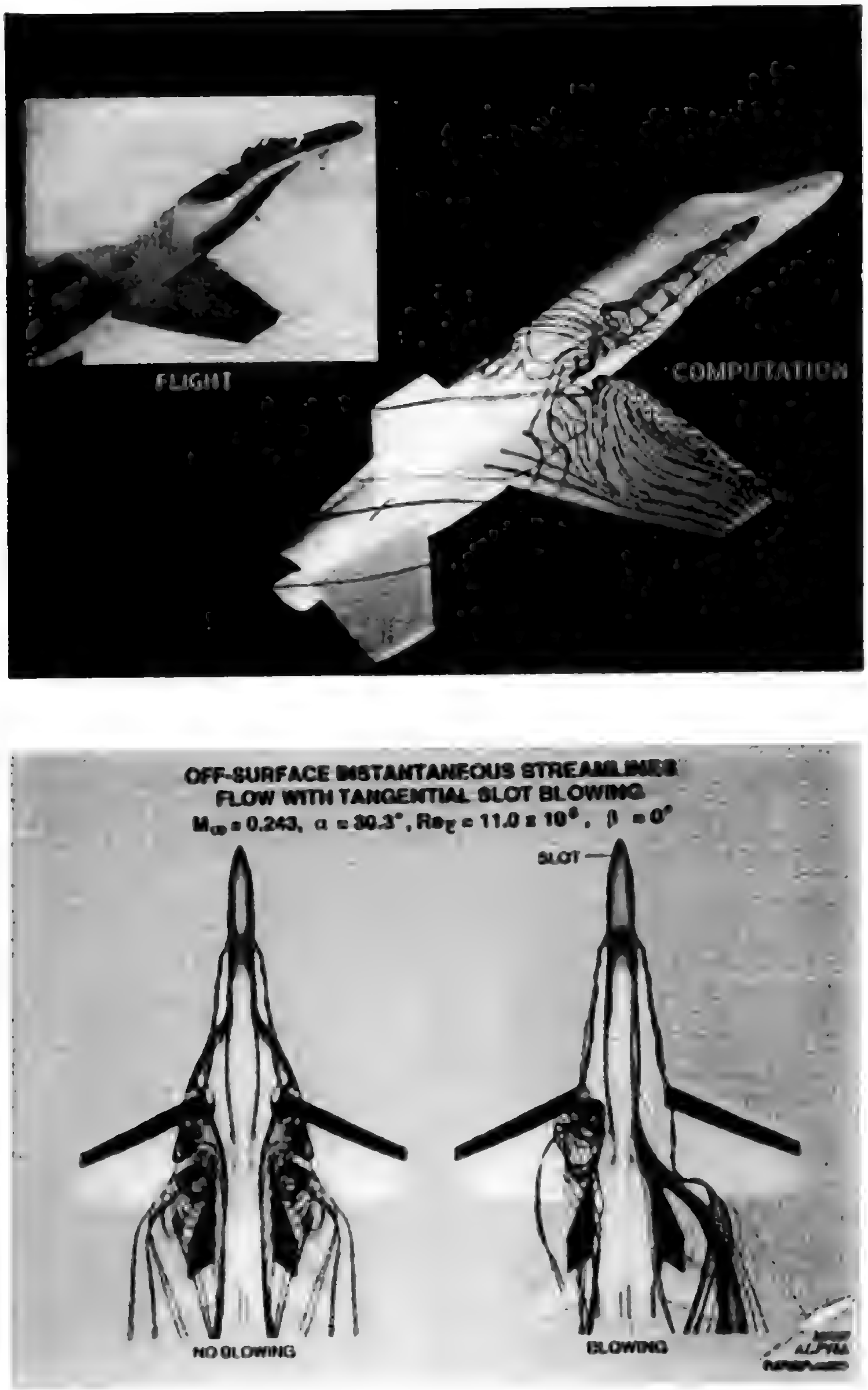
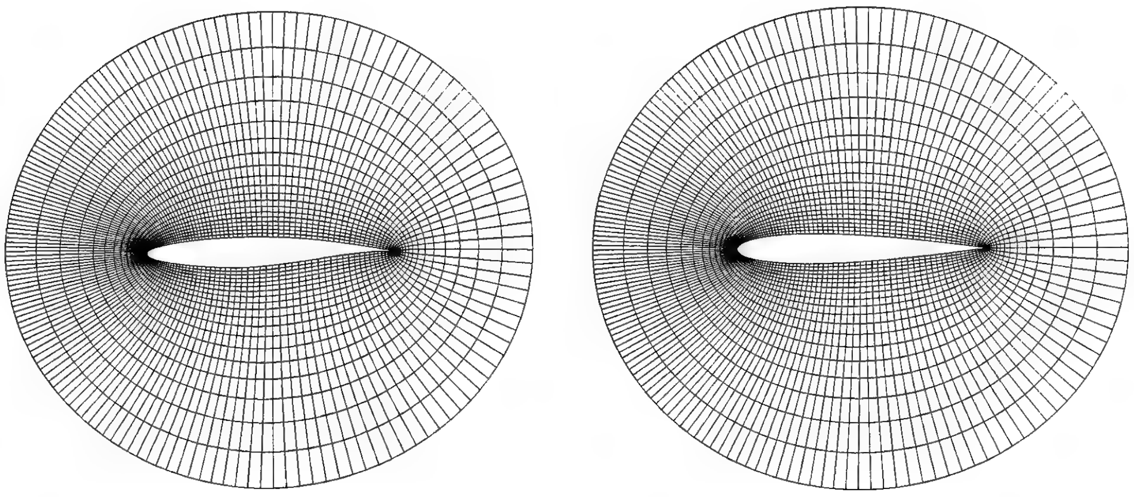

FLIGHT
COMPUTATION
OFF-SURFACE INSTANTANEOUS STREAMLINES
FLOW WITH TANGENTIAL SLOT BLOWING
$M_\infty = 0.243$, $\alpha = 30.3°$, $Re_{\bar{c}} = 11.0 \times 10^6$, $\beta = 0°$
SLOT
NO BLOWING
BLOWING

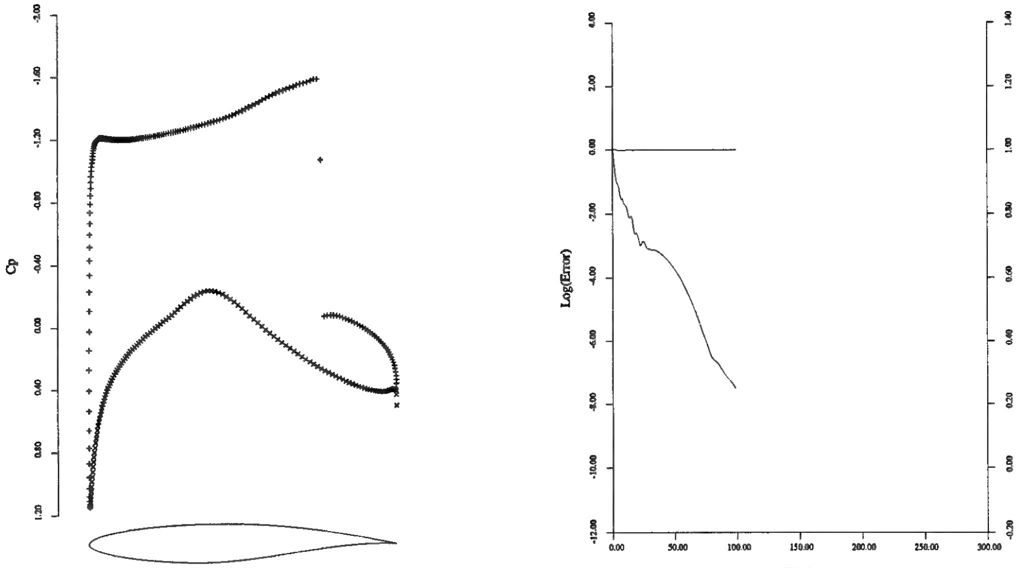
Figure 21: Navier-Stokes Predictions for the F-18 Wing-Fuselage at Large Incidence



22a: RAE-2822 Airfoil

22b: NACA-0012 Airfoil

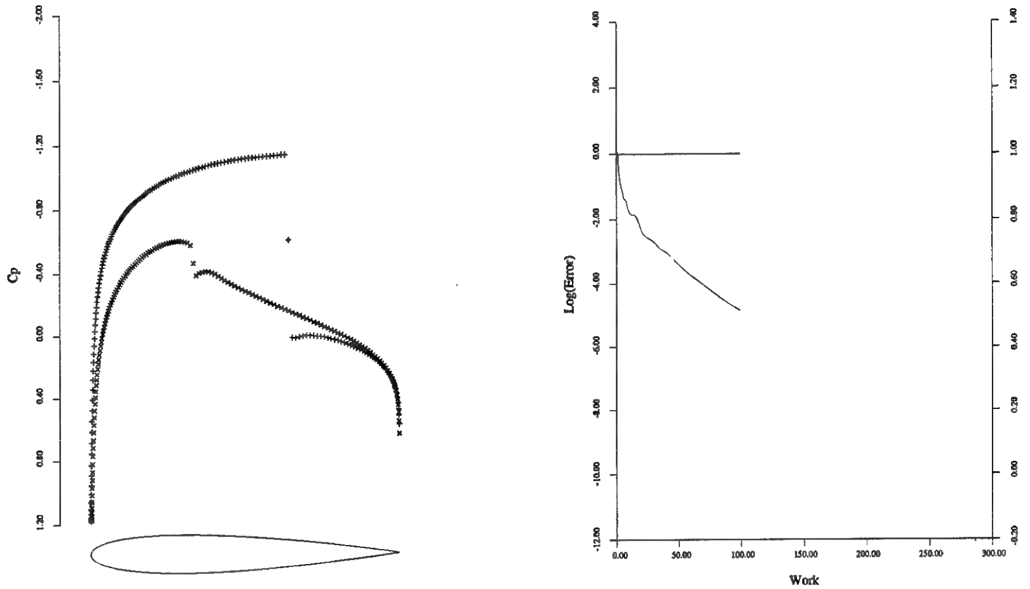
Figure 22: O-Topology Meshes, 160x32



23a: C_p after 25 Cycles.
C_l=1.1312, C_d=0.0469.

23b: Convergence.

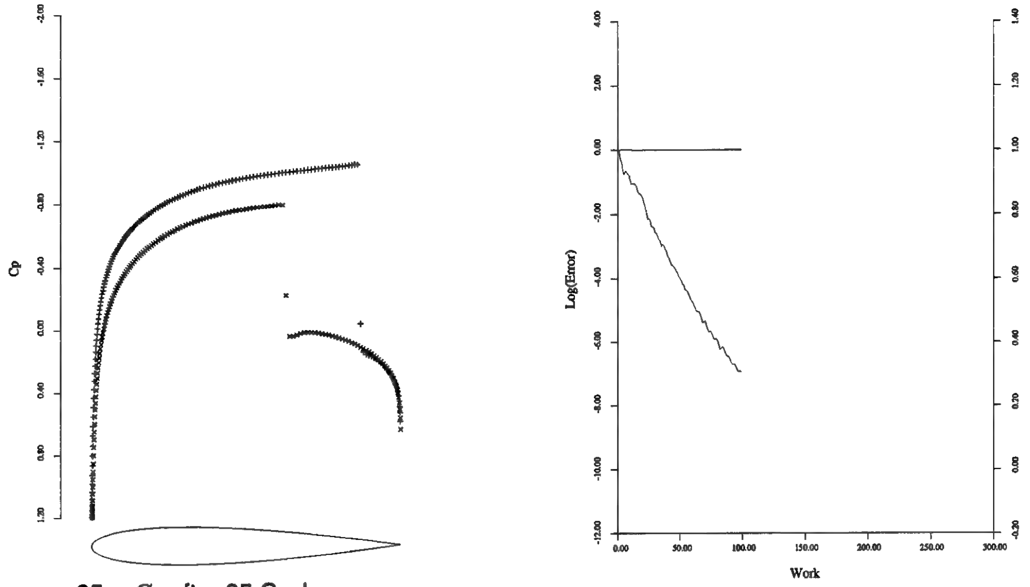
Figure 23: RAE-2822 Airfoil at Mach 0.750 and α=3.0°H-CUSP Scheme.



24a: C_p after 35 Cycles.
C_l=0.3654, C_d=0.0232.

24b: Convergence.

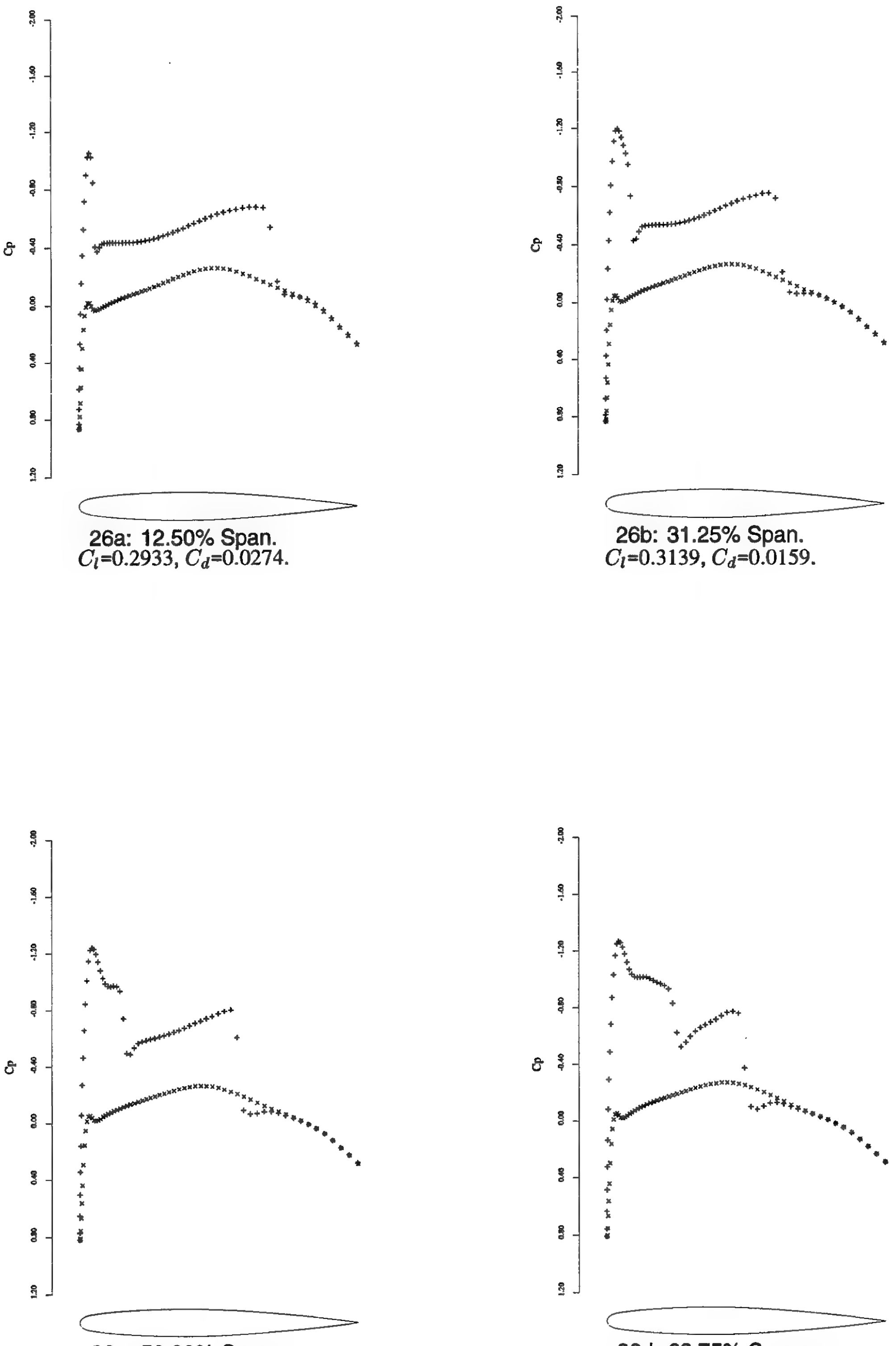
Figure 24: NACA-0012 Airfoil at Mach 0.800 and α=1.25°H-CUSP Scheme.



25a: C_p after 35 Cycles.
C_l=0.3861, C_d=0.0582.

25b: Convergence.

Figure 25: NACA-0012 Airfoil at Mach 0.850 and α=1.0°H-CUSP Scheme.



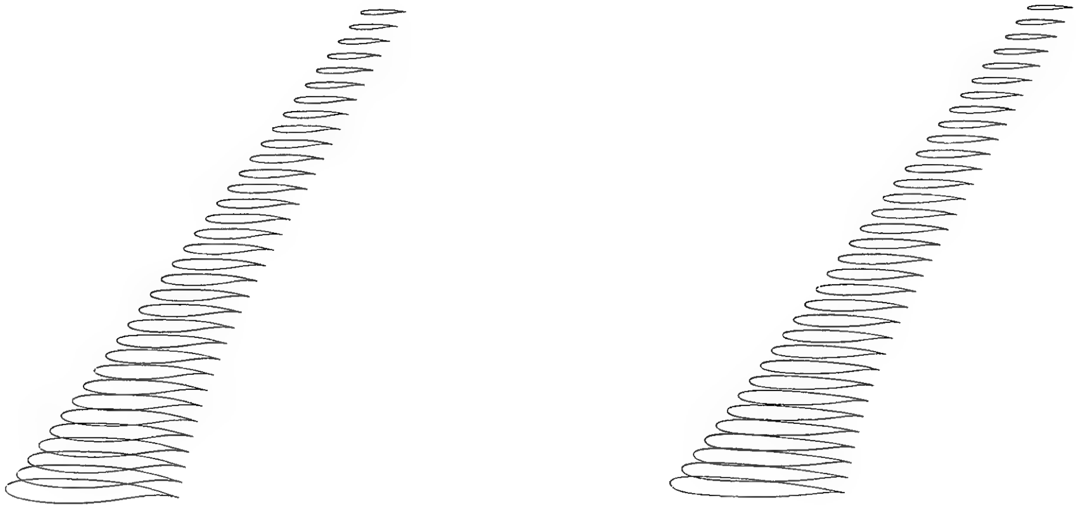
26a: 12.50% Span.
C_l=0.2933, C_d=0.0274.

26b: 31.25% Span.
C_l=0.3139, C_d=0.0159.

26c: 50.00% Span.
C_l=0.3262, C_d=0.0089.

26d: 68.75% Span.
C_l=0.3195, C_d=0.0026.

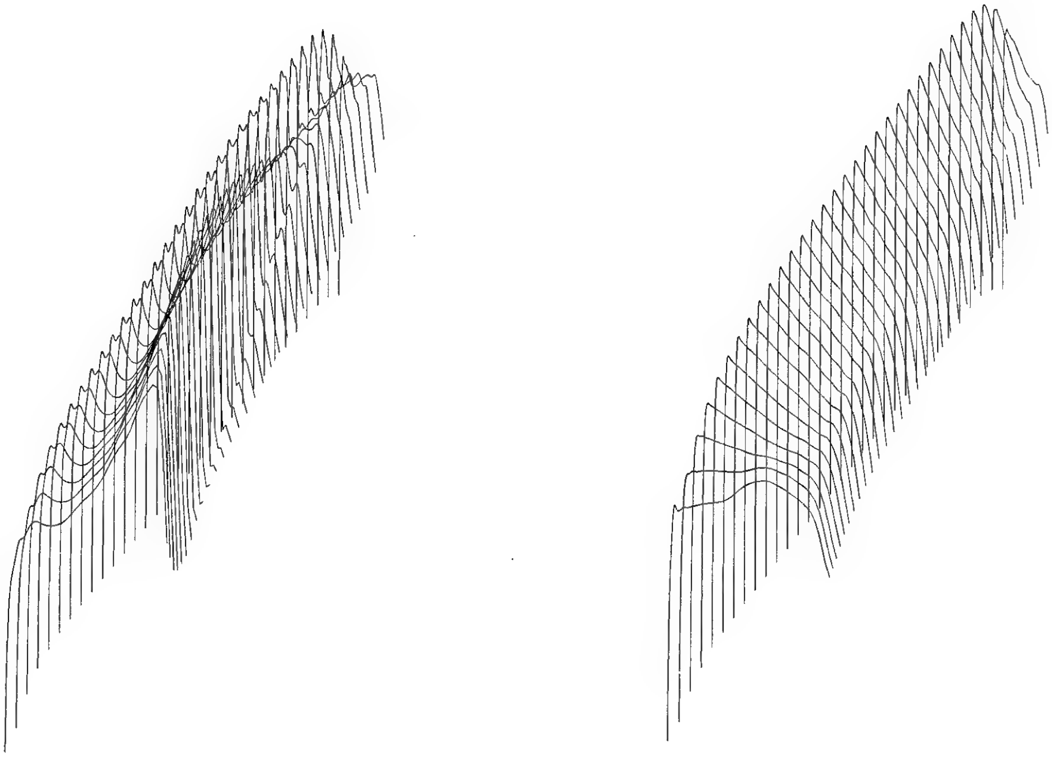
Figure 26: Onera M6 Wing. Mach 0.840, Angle of Attack 3.06°, 192×32×48 Mesh. C_L=0.3041, C_D=0.0131. H-CUSP scheme.



27a: Initial Wing
C_L=0.5001, C_D=0.0210, α=$-$1.672°

27b: 40 Design Iterations
C_L=0.5000, C_D=0.0112, α=$-$0.283°

Figure 27: Swept Wing Design Case (1), M=0.85, Fixed Lift Mode.Drag Reduction at C_L=.5.



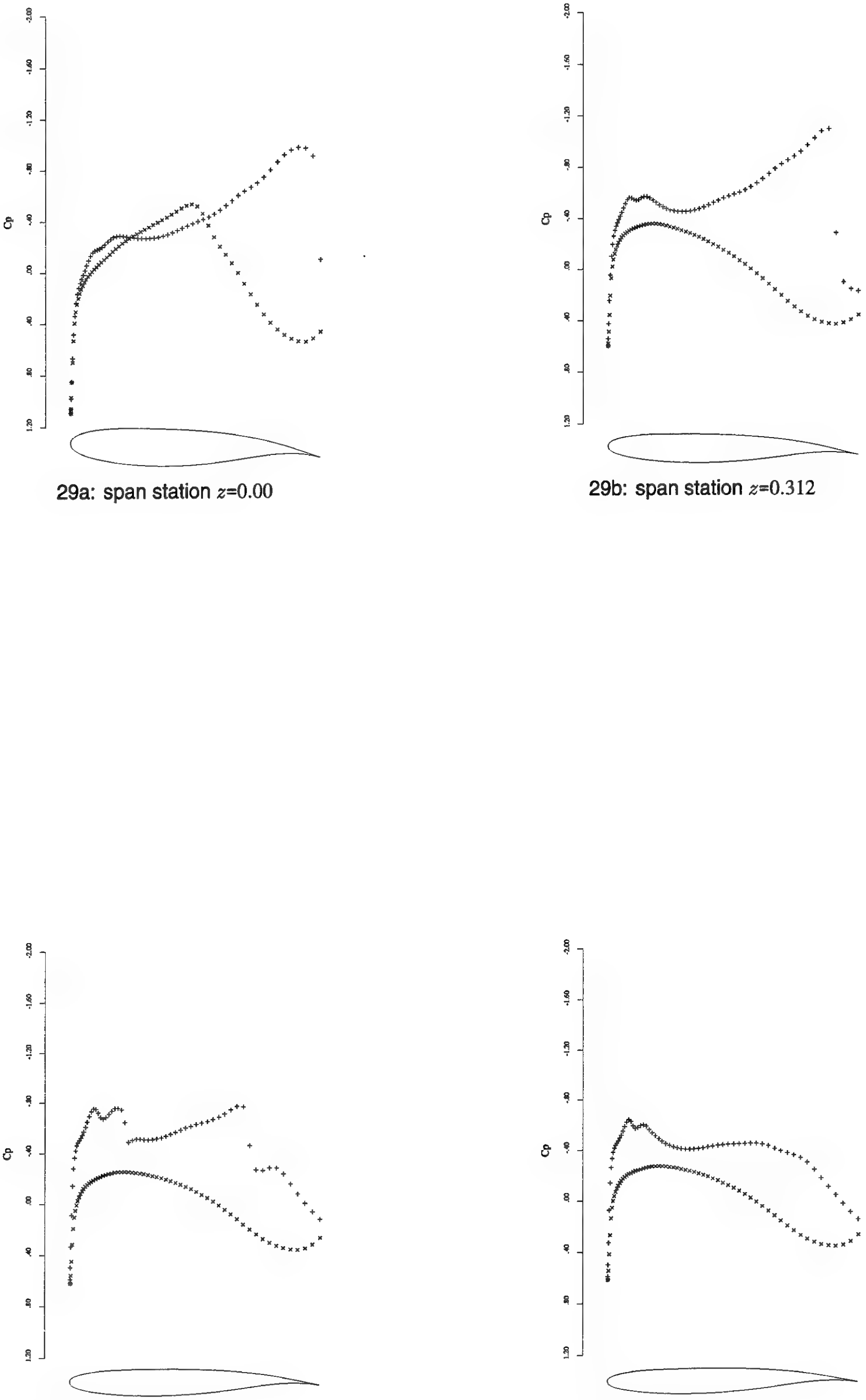
UPPER SURFACE PRESSURE

UPPER SURFACE PRESSURE

28a: Initial Wing
C_L=0.5001, C_D=0.0210, α=$-1.672°$

28b: 40 Design Iterations
C_L=0.5000, C_D=0.0112, α=$-0.283°$

Figure 28: Swept Wing Design Case (1), M=0.85, Fixed Lift Mode.Drag Reduction at C_L=0.50



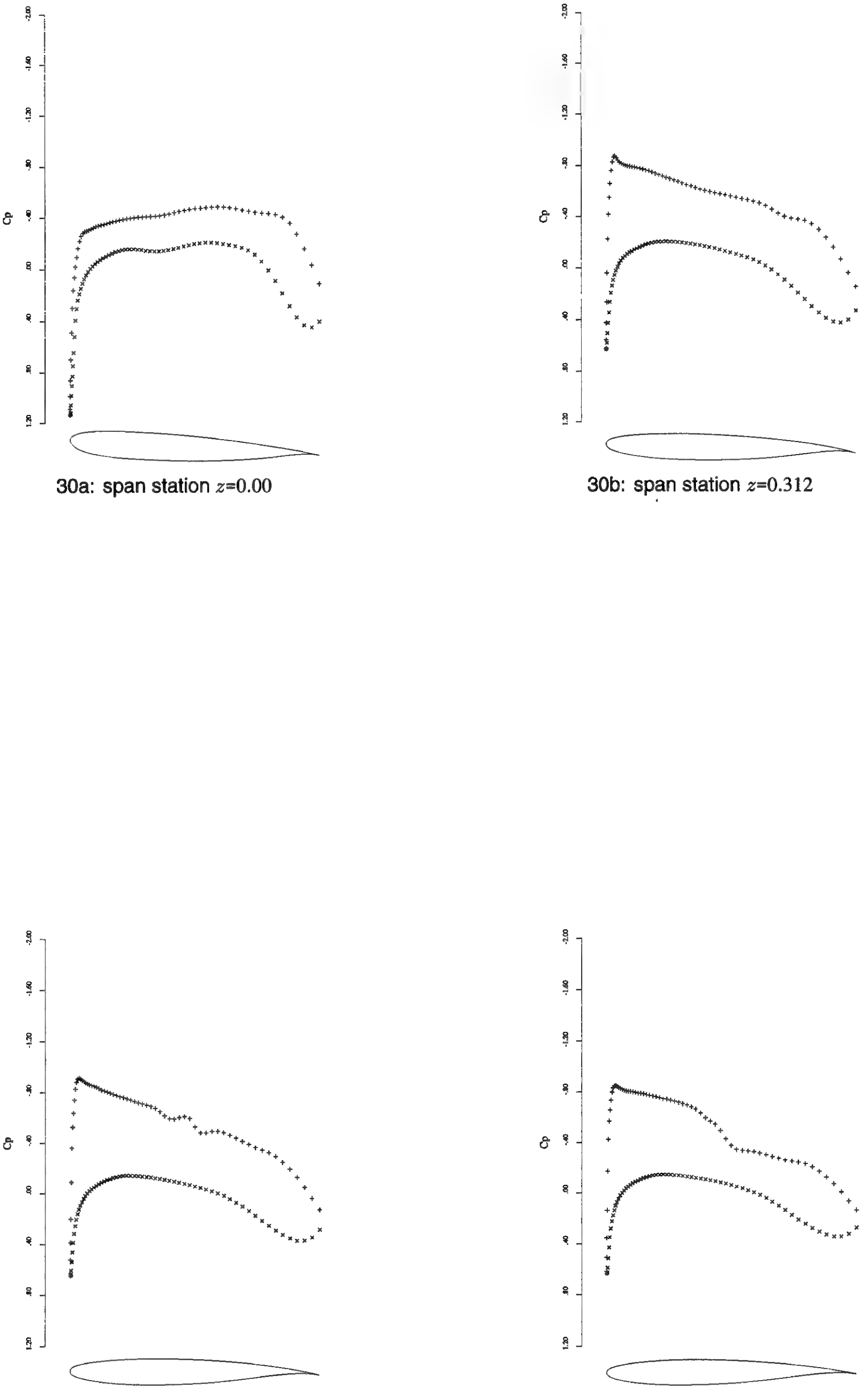
29a: span station z=0.00

29b: span station z=0.312

29c: span station z=0.625

29d: span station z=0.937

Figure 29: FLO67 solution for initial wing.M=0.85, C_L=0.4997, C_D=0.0207, α=$-1.970°$.



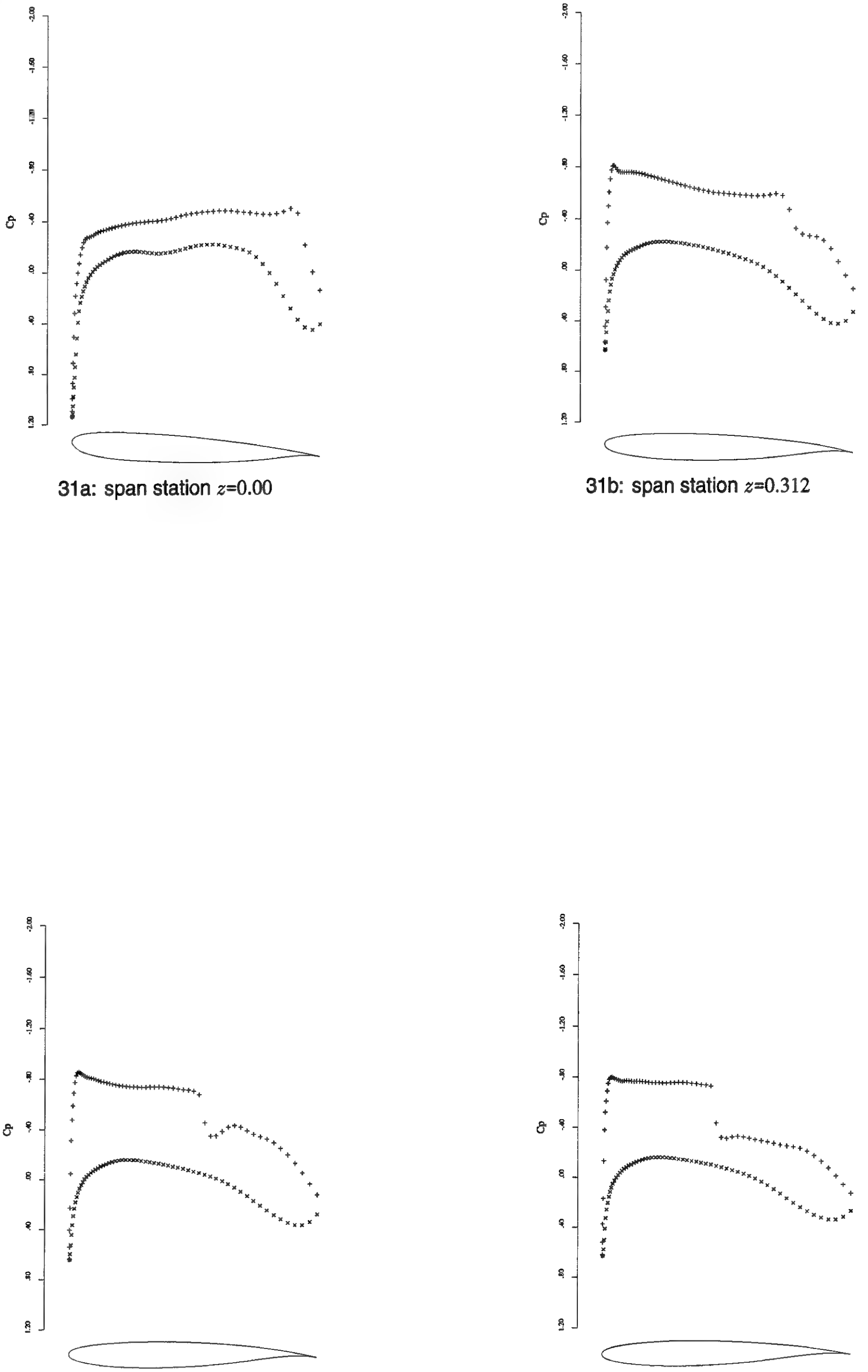
30a: span station z=0.00

30b: span station z=0.312

30c: span station z=0.625

30d: span station z=0.937

Figure 30: FLO67 check on redesigned wing. M=0.85, C_L=0.4992, C_D=0.0094, α=$-$0.300°.



31a: span station z=0.00

31b: span station z=0.312

31c: span station z=0.625

31d: span station z=0.937

Figure 31: FLO67 check on redesigned wing at a higher Mach number. M=0.86, C_L=0.4988, C_D=0.0097, α=−0.440°.

CFD RESEARCH IN THE CHANGING U.S. AERONAUTICAL INDUSTRY

Paul E. Rubbert
The Boeing Company
Boeing Commercial Airplane Group
P.O. Box 3707, M/S 67-UC
Seattle, WA 98124-2207
U.S.A.

SUMMARY

Changes are taking place in the world of CFD that extend beyond the technical. They include change to the "research engine," the system infrastructure that powers CFD research, as it seeks to adapt to the new industrial paradigm that is sweeping the aeronautics industry, and the world. The "research engine" involves government, academia, and industry. Because it is a system, all parts of it must participate in change. None of the parts can exist in isolation.

This paper analyzes the workings of the research engine and finds that it is encountering considerable strain. Resources for all elements of research are below historic levels. "Money givers" are faced with a lack of metrics and infrastructure for telling them how to invest their resources except in high level terms. Leaders of research are having to redefine their jobs. Researchers are hunkered down to wait it out. And value systems are in disarray and conflict. The adaptation of the research engine to the changing world is far from complete. It is in transition.

The paper goes on to describe what the author believes to be the principal characteristics and attributes of a well-functioning research engine, together with a few personal experiences that shed some light on how those attributes can be achieved. He concludes that further adaptation of the research engine will be paced by two key factors. One is the need to change the types of communication that take place between the research community and the engineering community in industry. The other is the need to unshackle the minds of researchers from the imprisonment of an overly narrow value system, a task which must be led by the money givers who inhabit the research engine.

INTRODUCTION

I find it interesting to contemplate those topics which are likely to be the pacing items and new challenges in CFD. Traditionally, such an endeavor would focus on the technology issues associated with CFD; things like algorithmic developments, hardware architectures, and so forth. Concerning the latter, Pradeep Raj has recently presented an up-to-date review (ref. 1) of the issues and pacing items in CFD technology. It addresses both the functional characteristics and the operational requirements that tomorrow's CFD codes must have in order to be effective, and it speaks for the U.S. aeronautical industry. Also, Professor Antony Jameson, the first keynote speaker, will share with us his vision of the technical challenges and future developments in CFD. There is little that I could add to their remarks. Therefore, I will focus my remarks on challenges and pacing items that extend beyond the technical.

We live at an interesting time. Our world is immersed in a period of large and rapid change. It is moving away from a dogmatic belief and reliance upon technology innovation as being the most significant element of competitiveness. That is being replaced by a new paradigm, one that is centered about customer satisfaction, quality and value as key goals.

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

We in industry are pursuing those goals by focusing on processes (ref. 2). We now understand that the key to developing better airplanes is to analyze, understand, and improve the processes by which airplanes are created. Similarly, the key to developing better CFD is to analyze, understand, and improve the processes by which CFD is created. We also now understand that the leading principle of good processes is customer focus and customer satisfaction. That principle applies equally to the processes that produce airplanes and the processes that produce CFD.

And so it seems to me that the most significant pacing item in the world of CFD is the need to analyze, understand and improve the process by which CFD capabilities are created. I call that process the research engine. There is more leverage in fixing up the research engine and adapting it to the changes in the world than in anything else I can think of. And so that is what I am going to talk about.

The "Old" Research Engine and How it Worked

The research engine as we know it today involves industry, academia, and government. Those three components interact with each other as a system. And like most systems, one component cannot be changed without affecting the others. It doesn't work for industry to change and the others not to change. We are all in this together.

The need to change pervades the entire research infrastructure. It involves information systems and the methods by which we communicate, including the holy grails of technical societies, publications, and technical conferences. It involves the changing of value systems, which is almost a cultural characteristic. And reward systems. Changing is not easy.

I would like to begin by examining how the research engine functioned in the era that we are leaving behind. Figure 1 (see Page 2-3) presents a description of the fundamental factors and forces that powered the research engine. This description appears to be universal. It looks the same, no matter whether you reside in industry, academia, or in a government laboratory. It works the same way. Only the names of the players may differ.

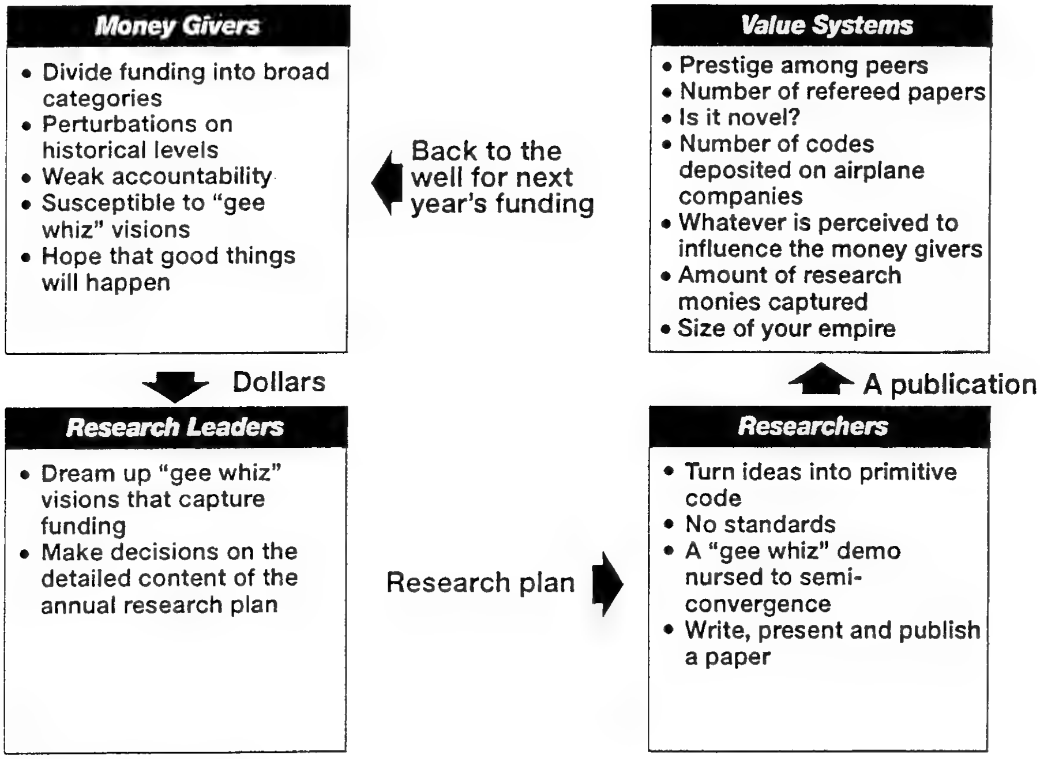
Key players are the money givers. Their role is to divide up money into various large buckets, each directed at a particular category of research, and to distribute it. We all know who those people are. They are the ones to whom we write research proposals. Money givers can be found in NASA, in the National Science Foundation, in the Department of Defense, and in similar institutions in Europe. They also are present in industry.

Most money givers are not close to the real details of airplane design, or to the detailed processes that use CFD as a tool. They operate at a higher, more strategic level. But they still need criteria by which to decide how to divide up the money. It is instructive to take a look at what some of those criteria were.

One such criteria was to divide up the money based on historical precedent. That was, and still is, practiced far and wide. It is a symptom of zero accountability and zero ability to discern what is important.

Money givers are also susceptible to being influenced by the visionary utterances of the people who inhabit the lower left box of figure 1, the research leaders. Research leaders are in the business of creating and marketing visions of how to make the world better. Many of them have become very good at creating visions for research that will be looked favorably upon by the money givers. *They treat the money giver as the customer.* One result of that, of course, is that the research funding decisions that get made can be quite unrelated to the true needs of the people who design airplanes for a living.

Money givers also are desirous of evaluating the caliber of the researchers to whom they will give money. It is rarely possible to



Money Givers

- Divide funding into broad categories
- Perturbations on historical levels
- Weak accountability
- Susceptible to "gee whiz" visions
- Hope that good things will happen

Back to the well for next year's funding

Value Systems

- Prestige among peers
- Number of refereed papers
- Is it novel?
- Number of codes deposited on airplane companies
- Whatever is perceived to influence the money givers
- Amount of research monies captured
- Size of your empire

Dollars

A publication

Research Leaders

- Dream up "gee whiz" visions that capture funding
- Make decisions on the detailed content of the annual research plan

Research plan

Researchers

- Turn ideas into primitive code
- No standards
- A "gee whiz" demo nursed to semi-convergence
- Write, present and publish a paper

Figure 1. The Engine That Powered CFD Research

point to a feature on an airplane and say "this research contributed to ---." So, it was necessary to establish other measures in order to create a value system which could be applied to individuals.

One popular measure was to look at the prestige bestowed upon a researcher, not by his customers, but by his peers, the other researchers. Can you imagine what automobiles would be like if the criteria for designing them was to please the other designers, rather than the people who want to use cars to drive about in?

Another popular measure has been to count the number of refereed papers that are produced by a researcher. One consequence of this is that our journals and conferences have become littered with papers whose real contribution is low or nonexistent. The journals have evolved into being primarily a scorekeeping system. Scientific information today travels largely by other means. Another consequence of the numbers game is that it encourages researchers to attack problems that they know how to solve rather than the problems that need to be solved. And so our entire research infrastructure was caught up in a value system that was largely unrelated to what was important to the engineers who design and build airplanes for a living. What counted was paying homage to a value system that controlled access to the annual pot of money necessary to support the research leader and his/her staff.

A standard part of the job of being a research group leader was also to make all of the important decisions concerning the detailed content of the annual research plan. After all, since research leaders are normally exposed to new and emerging technology that a design engineer is not, it was quite obvious to research leaders that they, and not design engineers, should be in charge of defining the annual research plan. And so the design engineering community was excluded from participating.

The researchers themselves focused their work on paying homage to the value system, because that is what entitled them to go back to the well for next year's funding, and to become eminent in the eyes of their peers.

So there it is. A stable, self-sustaining research engine that was capable of functioning quite smoothly, all by itself in its own little world. It did so for many years. Its weakness, of course, is that it had been almost disconnected from the community of people who we now understand to be the customers of CFD research, namely the practicing engineers who design airplanes for a living.

Figure 2 (see Page 2-5) exhibits the interfaces, such as they were, that existed between the research engine and the aeronautical industry. One such interface involved the money givers, who were visited periodically by clouds of collective wisdom passing overhead. Those clouds appeared in the form of high level advisory committees, wishes of the U.S. Congress, or of industry executives, depending upon where the money giver happened to reside. It is not entirely coincidental that these clouds are shown to be comprised of the condensation of hot air rising from airplane companies. In any event, the resulting fallout from these clouds caused the money givers to occasionally re-balance their research portfolios.

The other interface lay between the researchers and the practicing engineers who reside in airplane companies. This interface is characterized by the fence in figure 2. Interestingly, the site of the fence was not always in front of the door of the airplane company. It frequently could be found inside the airplane company, standing between the internal company research department and the practicing engineers who designed the airplanes. In those cases, the company research departments paid most allegiance to the research engine and acted as an integral part of it, particularly if they were dependent upon outside contract funding as a source of research money.

Communication over the fence was mostly one way. It consisted primarily of attempts by researchers to interest the engineering community in the products of their research. The system coined a name for this, calling it "technology transfer."

The favored means of lofting the results of CFD research over the fence was to send it across on the wings of a scientific publication. The publication was the messenger that told of its charms and attributes. And to make sure that at least some folks in the airplane company would see it, the researcher empowered his delivery system to honk, to attract attention. Such honking is frequently heard at technical conferences and symposia. In fact, that seems to have become the prime motivation for conference attendance. Overlooked was the fact that airplane design engineers rarely attended those conferences.

The boards of the fence have names inscribed upon them, entitled "conferences," "journals," "perceptions," "value systems," "reward systems," etc.. Those pillars of tradition and conventionality are turning out to be among the factors that impede our ability to create a research engine that is more properly connected to the customer.

It was a very eye-opening experience to us in the United States when NASA instituted some dramatic changes in communication. They changed the format of some of their conferences from one wherein the researchers did all of the honking to one in which industry did most of the talking and researchers did most of the listening. Lo and behold, it was discovered that the research community was not in fact immune to learning about what was important. We found that they could even learn from people who didn't have PhDs and a lengthy record of refereed publications. The power of two-way communication began to be unlocked!

So, somewhere along this journey of change we must abandon or at least supplement our old, one-way habits of communication as

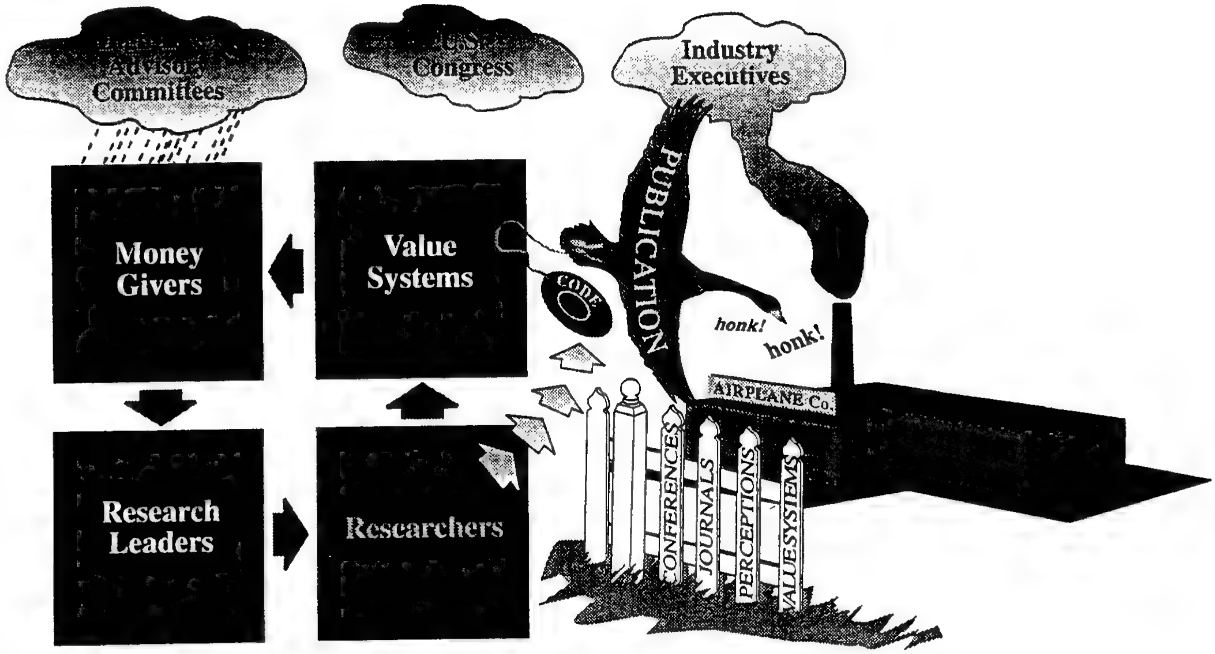


Figure 2. Interfaces With the Airplane Industry

institutionalized by the conventional scientific establishment. We must replace them with forms of communication that do the job that needs to be done. We are faced with the challenge of tearing down a fence whose pillars appear to be set in solid concrete.

<u>The Status of Change</u>

The process of changing has begun. Various people and organizations in all parts of the research engine are experimenting with new and different ways of operating. We are searching for a more effective research engine, but we have not found it yet.

When I look around, I see an increased level of tension throughout the infrastructure. Many researchers feel pressured to become more "applied." NASA is being battered from many sides, with some voices calling for them to get back to basic research while others are calling for them to increase their relevance to industry's needs. Academia is struggling to play a part, while finding a role for the individual graduate student and the educational mission. There are many conflicting forces at work. It is a difficult problem to even think about, much less resolve.

But we have changed. Figure 3 (see Page 2-6) presents my view of the current state of affairs in the United States. In the right hand part of the figure one can observe a new player appearing in industry. At Boeing we call these people "process owners," but that is not a universally used name. What is universal is the realization within airplane companies that processes are really important and that somebody must therefore be in charge of them.

And so, these "process owners" represent a new connection to the research engine. They have created a gap in the fence through which their voices are being increasingly heard. Money givers, research leaders, and researchers alike in government, academia,

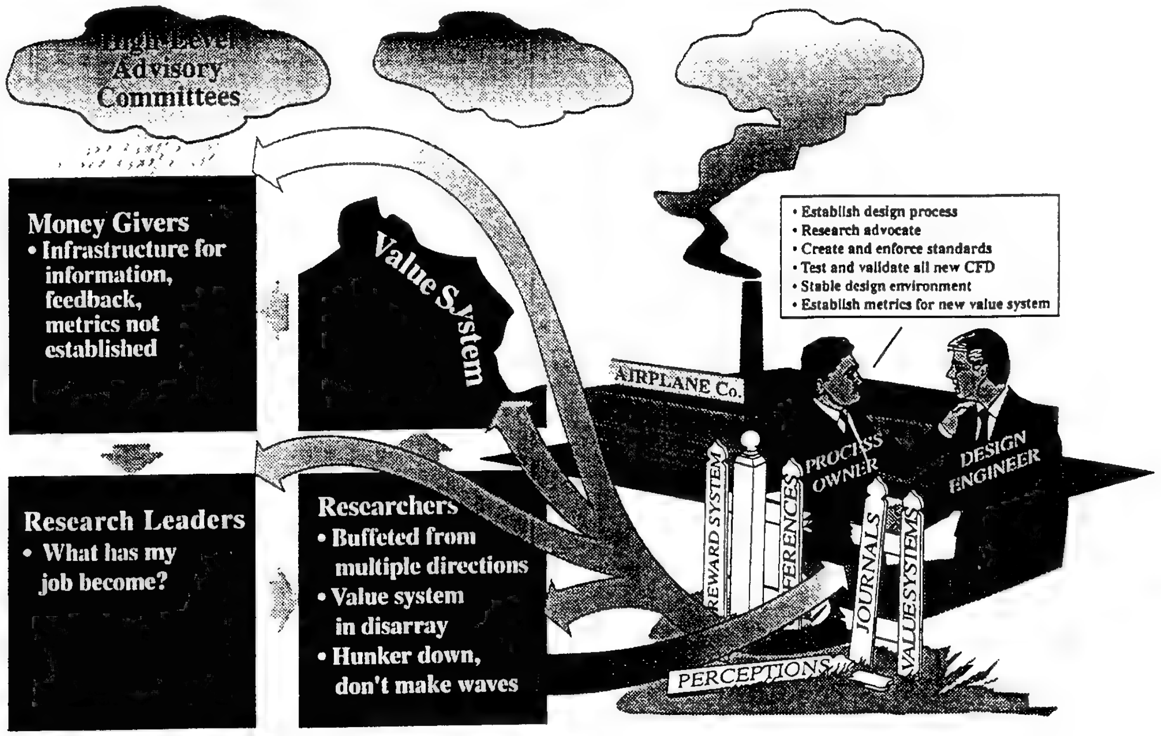

High-Level Advisory Committees
Money Givers
• Infrastructure for information, feedback, metrics not established
Value System
Research Leaders
• What has my job become?
Researchers
• Buffeted from multiple directions
• Value system in disarray
• Hunker down, don't make waves
• Establish design process
• Research advocate
• Create and enforce standards
• Test and validate all new CFD
• Stable design environment
• Establish metrics for new value system
AIRPLANE Co.
PROCESS OWNER
DESIGN ENGINEER
REWARD SYSTEM
FERENCES
JOURNALS
VALUESYSTEMS
PERCEPTIONS

Figure 3. The Research Engine in Transition

and industry are being exposed increasingly to their input. They are postured to evolve eventually into a strong component of the overall research engine.

The life of a money giver has become more challenging. They now all subscribe to the new vision of investing in things that reduce cycle time, cost, and so forth. But they mostly lack an infrastructure of established methods and metrics to guide them. They are inventing and innovating as they go.

The life of a research leader is also changing. The more progressive ones view their new role to be to *define and manage the process* of developing the annual research plan, rather than personally making the planning decisions. The new style of operating that I most frequently encounter is for the research leader and researchers to simply ask the process owners what they want and need, and then to set about implementing it. That is not leading to many of the attributes that we desire in the research engine.

The individual researchers are impacted by this evolving research engine and its changing power structure. They feel buffeted from several directions, not the least of which is a value system that is crumbling and in disarray. Their attitude is "don't make waves and do what people in power say they want done." They are not particularly happy.

And so we have not yet arrived at a properly functioning research engine. I don't have a complete vision of what that research engine would look like, but I do know many of the attributes that it should have. Those attributes include:

- a lead role in supporting the strategic direction set by the industrial enterprise.
- a proper balance between basic and applied research across the R&D food chain.
- a recognition of the importance of vision building within the research process.

- ability to draw upon all of our intellectual resources, both in developing the research plan and in executing it.
- nimbleness in translating the output of research into products and processes.
- a value system that causes more of the "right" things to occur.
- and perhaps most important of all, a value system that supplies high levels of human motivation and sense of worth, one that leads people from within to do more of the right things, and to make it fun once again to be a researcher.

Even though my vision of how to accomplish all of that is yet incomplete, I find within myself a growing conviction about some of the things that tomorrow's research engine must contain. One of those things is a better understanding of the proper distribution of roles, responsibilities, and core competencies that should prevail across the R&D food chain comprising industry, academia, and government. What that distribution should be can be derived by testing it against the axioms that accompany the new industrial paradigm, an exercise that certain segments of the research establishment find to be somewhat threatening. One outcome of that testing is the finding that a best and proper role for academia, and for much of NASA, is to concentrate on the foundational, overarching, enabling technology research which comprises the head of the R&D food chain.

Another of my convictions is that we must find a much better way of connecting the top and the bottom of the R&D food chain. This is something that we as a country have not yet learned to do well at all. And yet the issues involved are central to achieving a research engine that contains the attributes that we desire.

Connecting the two ends of the food chain is an issue in communication. We have to develop an understanding of what needs to be communicated, and when. And then we have to institute mechanisms to make it happen.

I have been fortunate enough to have enjoyed the privilege of running a research operation that encompassed the entire span of a research food chain, from foundational, enabling algorithm technology, and fluid mechanics, to production software, and customer support. In that position, I was able to experiment, so I learned about some things that don't work and other things that do work in properly connecting the two ends of the R&D food chain.

One thing that doesn't work well at all is to have research leaders at the head of the food chain simply ask the folks at the bottom of the food chain what they want or need, and then to blindly carry out their wishes. That leads mostly to short-term, evolutionary improvements of limited vision. It leads to tactical research rather than the strategic research which belongs at the head of the food chain, and it places the researchers in the position of "the boiler room" staff. They have much more to offer than that. Many people have yet to learn the true meaning of the words "customer focus" that have entered our language.

What does work, not only well but incredibly well in connecting the two ends of the R&D food chain, is to do the following four things:

1. Eliminate the constraints imposed in a researcher's mind by the value system under which he/she was educated. Make it O.K. to do things that are outside of the limits imposed by an overly narrow value system. Create a mind-set and a curiosity within the researcher to wander freely up and down the R&D food chain and even into manufacturing.

This is easier said than done. But it can be done. I've done it! It *has* to be done, because, more than anything else, ***it is the key that unlocks the power and the potential of the highly educated, highly***

paid research people whose minds have been refined and proven by our rigorously competitive academic system up to the PhD and post doctoral levels. The organization or the nation that does this earliest and best will have a very significant competitive advantage.

2. Expose and educate the researchers who inhabit the head of the food chain in the high level strategic thinking that supports the enterprise in which their customers are engaged. We don't do that very well today, and yet this is the element that enables researchers to identify and prioritize head-of-the-food-chain research topics in accordance with their strategic leverage. It must be realized that ***strategically relevant research should be the primary responsibility of the head of the research food chain***. The lower levels of the chain, where most process owners reside, are focused on tactical implementations.

3. Expose the researcher to the real world of the aerospace engineer. This only works well when carried out at the engineering site. Let the researcher "look over the shoulder" of the engineering community or the process owner as they encounter their daily challenges. Let the researcher build personal relationships with real engineers. What works even better is to expose a team of researchers encompassing a complementary set of differing skills and strengths, because you will then be deploying a more complete set of intellectual assets. The imperative is to *"enable the researcher to look beyond what the customer says he/she needs, and to formulate a vision of what he/she could provide that would really be useful to the customer and his/her environment."* This is not accomplished by talking mostly with the management, which has been our past practice. It is the direct exposure to the daily issues faced by the engineering design process that really turns on the creation juices. It is what enables the parable which says "necessity is the mother of invention" to operate!

These three steps, tearing down the imprisoning walls of the value system, exposing the researcher to the strategic thinking, and exposing him/her to the engineering world so as to enable the researcher to look beyond what the customer says he needs is *the one means that I have found to be consistently successful in creating ideas for research that have high relevancy and which are supercharged by bringing to bear the latest and greatest in enabling technology while bathed in the light of high level strategic thinking*. This is what we must strive for in our research engine.

The reason that one must proceed to a fourth step is that, at this stage, the customer will generally not agree with or approve of the ideas and plans that the researchers have formed as a result of the first three steps. Not yet!

The reasons are several. One is that the differing educational background of an engineer frequently makes it impossible for him to understand the approach being proposed by the researcher, or to assess the risk involved in turning ideas into reality. And engineers who are immersed in hot projects are much more focused on the answer they need tomorrow than in the strategic directions of interest at a higher corporate level. They tend to be tactical thinkers. But ***support and enablement of high level strategic direction is what head-of-the-food-chain research is all about!***

And so a means must be found to allow the researchers to proceed with development of their ideas in the face of customer opposition. This requires an act of courage on the part of the research leader and the money givers. But in my experience it rarely fails to produce handsome dividends. The only problem, if it should even be called a problem, is that at this stage nobody yet knows in exactly what form that dividend will be experienced. That appears only in step four.

Step 4 is what I call "vision building." This is the key activity that converts push to pull in the R&D food chain. The primary cause of failed research — and I define failed to mean research that doesn't get picked up and used by anybody — is that the vision from the head of the food chain that propelled the research, and the vision from the bottom of the food chain about what those folks think is useful, have no common intersection. If those two visions, originating from opposite ends of the food chain, cannot be made to intersect, the research will not be accepted. It will be ignored by the people who call the shots in determining what CFD gets used in the design of airplanes.

And so, a key element in the successful operation of an R&D food chain is the process that I call "vision building," a process for bringing together the separate visions that originate at the two ends of the R&D food chain. What does it take? Throwing publications or codes over the fence, which is the traditional approach to vision building, doesn't work well at all. Presenting "gee whiz" papers at conferences doesn't work. Arguing back and forth doesn't work well either. Neither does voting. I've tried them all.

What works is for the research community to produce something that an engineer can "touch and feel," usually a CFD code capable of performing a small number of computations that illustrate what can be done. This is not the time or the place for well-documented code, user friendly input formats, or polished and orderly software. Rather, the researcher at this point is engaged in a race to discovery and understanding before his fragile support system runs out of patience. Shortcuts are acceptable and encouraged, with one exception. That exception is execution efficiency. This is one of the key measures that will be "touched and felt" and usually should not be comprised.

Some people (managers and software specialists in particular) will be troubled with the idea of producing code that is undocumented, which probably does not adhere to standards, and which contains shortcuts. That is because they interpret the code to be the product. They fail to realize that ***the primary product of research at this stage is vision, not code!***

The best way I found to build vision was for the researchers to again return to the customer site. They would identify real design problems being faced by the design engineers and they would set up and run demonstrations of their new CFD technology on those problems. This led to side-by-side comparisons of new versus old ways of doing things. It frequently did not contribute much at that point to the engineering project's near term design goals because the code was still developmental, fragile, hard to use, perhaps containing a few bugs, and not yet trustworthy.

What it did do, and do well, was to build vision within the minds of design engineers. A typical reaction to a set of these calculations would be "so that is what you can do! Well, if you add this and that, I can use it for -----." That is vision building! At that point the engineer becomes an advocate of the research. ***This is when "push" changes to "pull" in the R&D food chain.***

The other thing that must happen is that the researcher must be able to now let go of his original vision, the one that led him to produce the CFD technology that is being demonstrated. He must allow himself to be influenced by the engineer-now-becoming-the-customer. He must adopt a new and better vision.

Vision building must be a two way street. It is a coming together, in the middle, of what were originally different visions at opposite ends of the food chain. It is not for one end of the food chain to convince the other end that its vision is best. ***It demands two-way communication. It is intense. It requires***

face-to-face interactions over a period of time. It demands that a new paradigm of communication be built into the research engine!

This is the type of vision building that generates the high levels of motivation and feelings of personal worth that must be present in a good research engine. It results in engineers and process owners anxiously awaiting the results of your research, calling you to find out how things are going, offering to help you, and telling your money giver to give you more money. It creates passion. It also causes researchers to drive themselves from within to work 16 hours a day, 7 days a week. In that kind of environment, it is a lot of fun to be a researcher.

Can we really be bold enough to think in terms of government or academic researchers really interacting with industry in those ways? Well, this past summer, I and the Director of ICASE (Institute for Computer Applications in Science and Engineering), Dr. M. Yousuff Hussaini, conducted an experiment in communication. He sent one of his research staff, Dr. Michael Lewis, to Boeing for seven weeks. One of those weeks was spent being tutored in the teachings of competitiveness and strategic direction. The other six were spent in learning and observing first hand what the practice of business acquisition, engineering, design, manufacturing, and customer support was all about. The thing that he was not allowed to do during these seven weeks was to engage in research.

At the end, I interviewed Michael. I found that he had learned enough to be able to "look beyond what industry says it needs and to gain an understanding of what he could contribute in terms of research that could really help industry but that we were probably unaware of." That is what we must strive to achieve in the minds of all research leaders who profess to be working at the head of the research food chain in areas that are related to aeronautics.

I cannot envision the entire population of university faculty and government laboratories descending upon industry sites for seven weeks each. But I can envision a selected subset of strategically placed research leaders perhaps doing it. And if we experiment with different formats and exposure times, we can probably reduce the exposure time significantly. *We simply must develop a new paradigm for communication!* Another interesting experiment would be to provide that type of exposure to the money givers who inhabit the research infrastructure.

I don't yet know what research Michael Lewis will choose to work on. That will be his decision. In any event, I am now contemplating a second experiment of inviting him back for a try at vision building whenever his research has progressed to the proper state. It will provide him with the opportunity to expose Boeing people to "touch and feel." I will attempt to measure his impact on the change in vision that he is able to create within Boeing people, and I will attempt to ascertain how his own vision has been caused to change. I will look for an intersection of those two visions as a measure of the effectiveness of his research.

In my view, the two purposes of communication that I am testing with the Michael Lewis experiments are ***the key*** communications that we must build into the research engine of tomorrow. ***One is to communicate strategic alignment and a broad understanding of the customer and his environment. The other is to provide a means for vision building, the process of achieving an intersection of the vision from the head of the research food chain with the vision from the engineering trenches. That is the process that converts push to pull and opens the door to industrial exploitation of research.***

The other component of the research engine that will be particularly influential in leading change is the value system. We simply must find a way to tear down the walls that are imprisoning the minds of many of our most brilliant people.

Value systems cannot be created or even modified very much by proclamation. It doesn't work to simply proclaim that we will now adhere to a new and different set of values. In the long run, it is the money givers within the research engine who have the only real power over the value systems. Value is ultimately associated with those endeavors that bring in money. That is true in industry, in academia, and in government. It will be up to the money givers to do the right things.

REFERENCES

1. Raj, P., "Requirements for Effective Use of CFD in Aerospace Design," NASA CP 3921, pp 15-28, May 1995

2. Rubbert, P. E., "AIAA Wright Brothers Lecture: CFD and the Changing World of Airplane Design," ICAS-94-0.2, September 1994

Parallel Computing in Computational Fluid Dynamics

Doyle D. Knight
Department of Mechanical and Aerospace Engineering
Rutgers University - The State University of New Jersey
PO Box 909 · Piscataway, NJ 08855-0909
knight@jove.rutgers.edu

Abstract

The paper presents an overview of parallel computing in computational fluid dynamics. A taxonomy of parallel computing architectures and programming paradigms is described. Issues in parallel computing are discussed including domain decomposition and load balancing, performance, scalability, benchmarks and portability. Examples of experience with parallel computing in the aerospace industry is described.

1 Overview

This paper is intended for researchers in Computational Fluid Dynamics (CFD) who do not have experience in parallel computing. It provides a description of parallel computing hardware architecture, software paradigms, the principal issues in utilizing parallel computing for CFD, and examples of use of parallel computing in the aerospace industry.

Parallel computing, particularly in computational fluid dynamics, is a broad field of research and development. The software and hardware technology is developing at an extraordinary pace. The reader is directed to the numerous journals on parallel computing (*e.g.*, *Inter. Journal of High Speed Computing*, *The Journal of Supercomputing*, *Inter. Journal of Parallel Programming*, *Inter. Journal of Supercomputer Applications*), as well as recent conferences and workshops (*e.g.*, Parallel CFD '95), for further information. Additionally, extensive information is available on the World Wide Web, *e.g.*, http://www.cnb.compunet.de/para/para.html, http://www.netlib.org/nhse/.

2 What is Parallel Computing?

This section presents an anecdotal discussion of the earliest refernce to parallel computing, describes Flynn's and Bell's classifications of parallel computer architectures, and briefly discusses the message passing and data parallel programming paradigms.

2.1 Introduction

Parallel computing is the simulataneous operation of multiple computational tasks on a computer system. Parallel computing has been an integral part of computing systems from their beginning. The earliest reference to parallel computing appears to be the description by L. Menabrea of Charles Babbage's computer. Among the principal virtues of an earlier (but evidently not final) design, Menabrea describes the capability (and importance) of parallel computing [1]:

> "...Secondly, the economy of time: to convince ourselves of this, we need only consider that the multiplication of two numbers, consisting each of twenty figures, requires at the very utmost three minutes. Likewise, when a long series of identical computations is to be performed, such as those required for the formation of numerical tables, the machine can be brought into play so as to give several results at the same time, which will greatly abridge the whole amount of the processes."

Also, the first general purpose electonic digital computer ENIAC, built to compute projectile and firing tables for the US Army in World

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

War II, was a parallel computer with 25 independent computing units (20 accumulators, 1 multiplier, 1 divider/square rooter, and 3 table look-up units) performing different tasks for the solution of the specific problem. Moreover, the ENIAC used decimal arithmetic internally (as opposed to the binary arithmetic used on modern computers) and operated on all ten decimal digits of a number in parallel. The ENIAC was programmed in hardware, *i.e.*, using a plugboard to wire connections between the units. However, the parallel computing capability of the ENIAC was never fully realized in practice. After two years of operation, it was reconfigured as a serial centralized computer [2].

There are four distinct **levels of parallelism** [1]. The highest level is *job*, where the computer system operates simultaneously on unrelated tasks (*e.g.*, a CFD simulation for an F-18 and a CEM simulation for a B-2). The second level is *program*, where the computer system operates simulaneously on different parts of the same program (*e.g.*, the parallelization of a DO loop across multiple processors). The third level is *instruction*, where the different instructions are performed in parallel (*i.e.*, fetching one instruction from memory while performing an arithmetic operation). The fourth level is *arithmetic and bit*, where parallelism is achieved within an individual arithmetic or bit instruction. This paper focuses on the second level (*program*) of parallelism in computational fluid dynamics. We consider the issues of parallelism in the context of a *single* program (*e.g.*, the simulation of a combustion chamber) operating on a parallel computer.

2.2 Classification of Parallel Computer Architectures

Flynn [3] originated a classification of **parallel architectures** which has become widely accepted (Table 1). Four distinct categories are defined based on the *data stream* which is the sequence of instructions and/or data executed or operated on by a processor. *Single Instruction Stream/Single Data Stream (SISD)* is the conventional serial architecture employing a single stream of data and a single processor. This is also known as the von Neumann computer (or architecture) or a serial computer. Modern single-processor workstations or micro-computers are examples of this category. *Single Instruction Stream/Multiple Data Stream (SIMD)* computers have several computational units which can perform the same operation (*e.g.*, adding two numbers) simultaneously on different parts of the data stream. An example is the Cray C-90. *Multiple Instruction Stream/Single Data Stream (MISD)* implies simultaneous different operations by separate computational units on the same data stream. Examples of this type are rare. *Multiple Instruction Stream/Multiple Data Stream (MIMD)* indicates multiple computational units operating simultaneously on multiple data streams. Examples are the Thinking Machines Corporation CM-5, the Cray T3D and, indeed, the ENIAC.

Table 1: Flynn's Taxonomy

Acronym	*Definition*
SISD	Single Instruction Stream - Single Data Stream
SIMD	Single Instruction Stream - Multiple Data Stream
MISD	Multiple Instruction Stream - Single Data Stream
MIMD	Multiple Instruction Stream - Multiple Data Stream

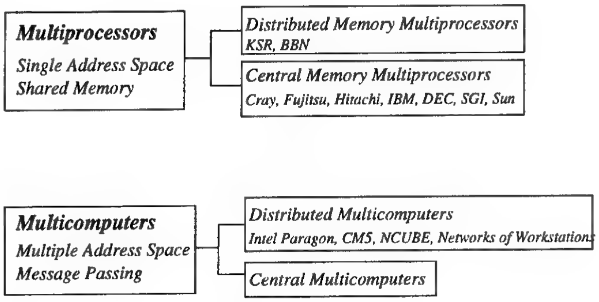
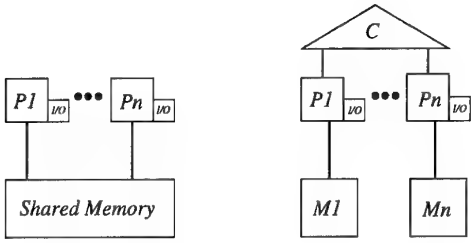

Multiprocessors
Single Address Space
Shared Memory
Distributed Memory Multiprocessors
KSR, BBN
Central Memory Multiprocessors
Cray, Fujitsu, Hitachi, IBM, DEC, SGI, Sun
Multicomputers
Multiple Address Space
Message Passing
Distributed Multicomputers
Intel Paragon, CM5, NCUBE, Networks of Workstations
Central Multicomputers

Figure 1: Bell's taxonomy of MIMD architectures (with examples)

Flynn's classification, although useful for broadly categorizing parallel computers and widely cited, is nonetheless incomplete, and various other classifications have been introduced. Bell [4] subdivides the MIMD category into two subcategories as indicated in Fig. 1. *Multiprocessors* are parallel computers with a single address memory (**shared memory**), *i.e.*, the central memory (RAM) is organized into a sin-



a) Multiprocessor

b) Multicomputer

Figure 2: Multiprocessor and Multicomputer

gle logical address domain which is accessible to all of the processors (Fig. 2). Processors $P1, \ldots, Pn$ can access the same data in memory (*i.e.*, the same address location), albeit not simultaneously. Examples are the Cray C-90 and SGI Power Challenge XL. *Multicomputers* are parallel computers with multiple distributed memory address spaces (Fig. 2). Processor $P1, \ldots, Pn$ have dedicated, independent memories $M1, \ldots, Mn$ which are not directly accessible by each other. Examples are the Intel Paragon, IBM SP2 and networks of individual workstations. If processor $P1$ needs to access data in the memory assigned to processor Pn, it sends a message to Pn requesting the data, and Pn complies. The transfer of data from the memory of one processor to the memory of another is denoted **message passing**, and is a principal characteristic of multi-computers. All communications between processors occur through a communications network C in Fig. 2. Many different types of communications network topologies have been developed (see Fig. 3 of Bell [4]).

The relative advantages and disadvantages of multi-processors *vs.* multi-computers have been widely studied, and numerous research (and production) machines of both types have been constructed [4]. Although greatly oversimplified, the main issues are as follows. For a multiprocessor, the shared memory eliminates the computational cost and program complexity of message passing. However, a multi-processor with a single shared memory is not scalable, *i.e.*, the architecture cannot simply be scaled to an arbitrary number of processors and arbitrary memory size. This arises from the limitation on data transfer rate (**bandwidth**) between memory and processors. This has led to a subdivision of multiprocessors into two categories, the *central memory multi-processors* as described previously, and the *distributed memory multi-processors* (Fig. 1) where the independence of the distributed memories is hidden from the user by means of an automatic data transfer mechanism (caching). For a multi-computer, the distributed memory eliminates the scalability problem associated with a single memory of limited bandwith. However, multi-computers incur the computational cost and program complexity of message passing.

Other classifications of parallel computers have been developed, *e.g.*, Shore [5], and Hockney and Jesshope [1].

2.3 Parallel Programming

There are two basic types of parallel programming paradigms (or environments). As the name suggests, **message passing** involves the explicit use of send and receive functions by the applications programmer. These functions communicate information between the memory assigned to individual processors. Many manufacturers of distributed memory parallel computers have developed specialized message passing libraries (*e.g.*, nCUBE, Intel), although standards are emerging (see §3.5.2 and 3.5.3). The **data parallel** paradigm involves a single program which controls the distribution of data across all processors, and the operations on the data. Typically, the data parallel language supports array operations and permits entire arrays to be used in expressions. Manufacturers of shared memory parallel computers have developed specialized compiler directives for data parallel programming (*e.g.*, Cray C-90 and SGI). An emerging standard for a data parallel language is High Performance Fortran (§3.5.1).

2.4 Examples of Parallel Computers

Table 2 lists a number of current parallel computers. It should be emphasized that the information shown does not fully describe the capabilities (and limitations) of a parallel computer. Other relevant factors include memory bandwidth, cache memory, I/O bandwidth, compiler technology, debugging software, etc. Furthermore, the performance specifications change frequently due to product upgrades.

Table 2: Examples of Current and Future Parallel Computers

Name	*Class*	*Max No. of Processors*	*MFlops/ Processor*	*Max Memory (GByte)*	*Type of Memory*
Convex SPP1200/XA	MIMD	128	240	32	Shared
Cray J-90	SIMD	32	200	8	Shared
Cray C-90	SIMD	16	1000	2	Shared
Cray T-90	SIMD	32	2000	8	Shared
Cray T3D	MIMD	2048	150	128	Distributed
Cray T3E (2Q96)	MIMD	2048	600	1024	Distributed
DEC 8400 5/300	SIMD	12	600	14	Shared
Fujitsu VPP300	SIMD	16	2200	32	Distributed
IBM SP-2	MIMD	128	266	256	Distributed
Intel Paragon XP/S 35	MIMD	512	150	16	Distributed
NCUBE-2	MIMD	4000	4	250	Distributed
NCUBE-3 (Dec 95)	MIMD	12000	100	3000	Distributed
SGI Power Challenge XL	SIMD	18	360	16	Shared
Thinking Machines CM-5	MIMD	512	160	64	Distributed
Thinking Machines CM-500 (Fall 95)	MIMD	2048	160	256	Distributed

LEGEND

GByte	Gigabyte (10^9 byte)
MFlops	Millions of floating point operations per second (theoretical maximum)

NOTES

1. *Maximum Number of Processors* may refer to processing *elements* on some systems.
2. *Memory* does not include secondary memory storage (*e.g.*, Solid-State Storage Device (SSD) on the Cray C-90/T-90).
3. Dates in parentheses indicate manufacturer's published date for availability.

3 Issues in Parallel Computing

Effective utilization of parallel computing in computational fluid dynamics involves numerous issues which must be adequately addressed. In this section, we focus on several key questions, in the context of development of new codes for parallel computing.

3.1 Domain Decomposition and Load Balancing

The partitioning of data and computational tasks among multiple processors is denoted **domain decomposition**. An example is shown in Fig. 3. A two-dimensional structured grid for a jet engine nozzle is partitioned into subdomains, and each subdomain assigned to an individual processor. This approach typifies the domain decomposition for a multi-computer. The domain decomposition may occur prior to or during the execution of the flow code.

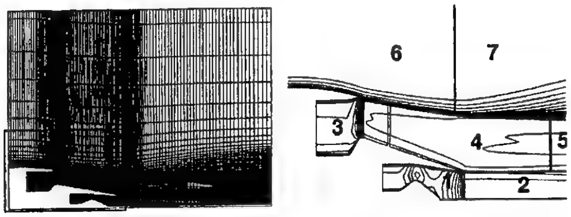


Figure 3: Multi-block grid (from [46])

The principal objective of domain decomposition is to maintain uniform computational activity on all processors. This is known as **load balancing**. For a fixed numerical algorithm (*e.g.*, the Euler equations) on a fixed grid, load balancing is straightforward, *i.e.*, each processor is assigned approximately the same number of cells. However, several factors can complicate load balancing. First, the nature of the governing equations can change during the computation. An example is combustion, where the chemical reaction source terms are computed only when the local static temperature exceeds a preset value [6, 7]. Second, the number of governing equations in a given subdomain can change. An example is particle tracking where particles can accumulate in a subregion (*e.g.*, recirculation zone). Third, the grid can change during the computation due to adaptation. Thus, in many cases it is necessary to incorporate dynamic load balancing, wherein the load on each processor is monitored and the overall task load redistributed to achieve an approximate uniform load.

An example of a simple dynamic load balancing method is presented in Borrelli [6] for hypersonic reacting flow. The chemical reactions are important only when the local static temperature exceeds 2000 deg K, and the ratio of computational work for reacting *vs.* non-reacting flow is approximately ten. The dynamic load balancing algorithm decomposes the domain by assigning a weighting function of either 1 or 10 to each cell, corresponding to non-reacting and reacting, respectively, and subdividing the domain to achieve an approximate uniform average weighting function for each subdomain.

3.2 Performance

A key issue is the performance of a CFD code on a parallel computer. Many different measures of performance have been proposed, and there is an active debate regarding the most appropriate. However, in solving a given problem (*e.g.*, viscous flow past an F-18), the true measure of performance is simply the **wall clock time to completion**. Inotherwords, given the opportunity to choose among different computational resources, the individual typically selects the resource which yields the answer in the shortest elapsed time, subject to existing constraints (*e.g.*, budget, system load, etc).

Of course, it is impossible to model this selection process in a universal manner, and thus the development of performance measures have focused principally on more ideal cases. One performance measure is **megaflop** (*i.e.*, millions of floating point operations per second) *vs.* number of processors. An example is presented in Fig. 4 from Simon *et al* [8] for two different codes: a 2-D unstructured Euler code [9], and a 3-D particle simulation code for rarefied gas flows [10]. Both codes were executed on an Intel iPSC/860 multicomputer for 2^n processors where $n = 1, \ldots, 7$. The unstructured Euler code achieves a substantially higher megaflop performance than the particle code.

Another performance measure is **efficiency**,

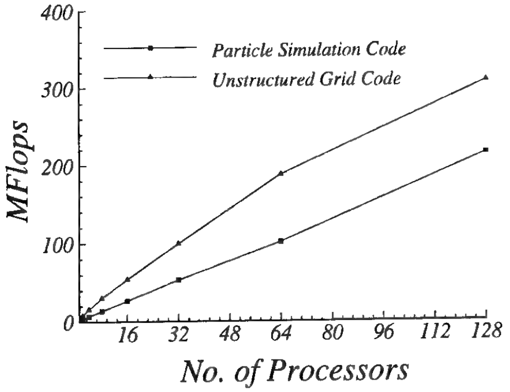


Figure 4: Megaflops of Two Codes on the Intel iPSC/860

i.e., the fraction of the peak performance (relative to a single processor) achieved on a machine by a specific code. It is defined as

$$\eta = \frac{\text{CPU time for one processor}}{\text{n} \times \text{CPU time for } n \text{ processors}} \qquad (1)$$

Typically, the efficiency η is plotted against the number of processors n. A related quantity is the **speedup** S defined as

$$S = n\eta \qquad (2)$$

Efficiency can depend strongly on the algorithm. An example is presented in Fig. 5 from Simon *et al* [8] for the same codes as in Fig. 4. Here the trend is opposite to the megaflop performance measure, *i.e.*, the 3-D particle code retains 88% efficiency at $n = 128$, while the 3-D unstructured code drops to 52% at $n = 128$.

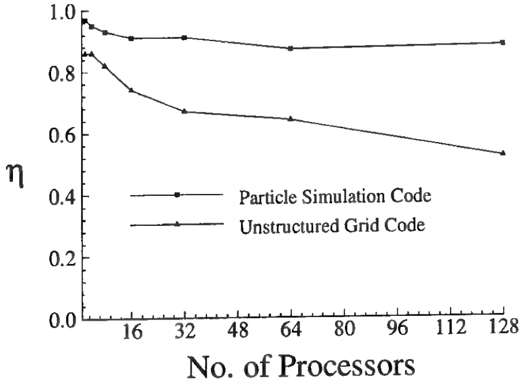


Figure 5: Efficiency of Two Codes on the Intel iPSC/860

3.3 Scalability

The impact of scaling a given parallel computer architecture to increasingly larger number of processors is a key concern. Although this problem may be viewed from several perspectives, it is instructive to examine it in the following context. Consider the solution of a given computational fluid dynamics problem, *e.g.*, a Reynolds-averaged Navier-Stokes simulation of an entire aircraft configuration using a fixed number of grid points. How does the *efficiency* of the computation depend on the number of processors ? This question may be treated (albeit simplistically) by a straightforward analysis proposed by Amdahl [11]. Denote the execution time of the program on a single processor by t_1. Assume that an analysis of the program and algorithm structure indicated that a portion of the code could be reprogrammed for parallel execution (*e.g.*, the product of a matrix and a vector, which is a common operation in iterative methods for solution of linear systems). Let t_p denote the cpu time on the single processor for this potentially parallelizable section. Let t_s denote the cpu time for the remaining unparallelizable (*i.e.*, scalar) code. Neglecting the cost of scheduling processors, communications between processors (if any) and synchronization time (*i.e.*, the time required to allow all processors to reach a common point following execution of the parallel section of code), the efficiency of a parallel computation with n processors is

$$\eta = \frac{t_1}{nt_n} = \frac{t_s + t_p}{n(t_s + t_p/n)} \qquad (3)$$

and defining the **parallelizable fraction** $f \equiv t_p/(t_s + t_p)$,

$$\eta = \frac{1}{f + n(1-f)} \qquad (4)$$

This is known as **Amdahl's Law** and is displayed in Fig. 6. The precipitous drop in efficiency for all but the highest possible parallelizable fractions is strikingly clear. Even for $f = 0.99$, the efficiency η is 0.5 at $n = 101$.

In some cases, the communications cost may yield even lower efficiencies than predicted by Amdahl's Law. Consider a fixed domain $\mathcal{D}$ of N^3 cells on a multi-computer with n processors. Assume a equi-distribution $\mathcal{D}_k$, $k = 1, \ldots, n$ of N^3/n cells to each processor. Typically, a **halo**

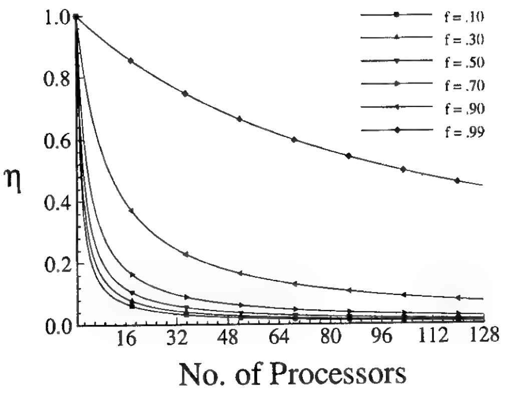


Figure 6: Amdahl's Law

of fictitious cells are added to each processor which represent the additional information necessary to integrate the flow variables within cells assigned to the processor by a single time step. The number of halo cells is proportional to the number of cells in $\mathcal{D}_k$ which share one or more faces with other subdomains $\mathcal{D}_l$, $l \neq k$, and is therefore $\mathcal{O}(N^3/n)^{2/3}$. The ratio of communications time to flowfield integration time, denoted by ζ, is therefore

$$\zeta \approx \frac{\mathcal{O}(N^3/n)^{2/3}}{N^3/n} = \mathcal{O}\left(\frac{1}{N^3/n}\right)^{1/3} \qquad (5)$$

Thus, for a fixed number of cells, the relative cost of communications can increase as the number of processors increases[1].

3.4 Benchmarks

Numerous benchmarks have been developed for parallel computers[2]. All benchmarks have limitations, of course, and the overemphasis on (and misuse of) benchmarks has naturally led to a somewhat skeptical attitude towards them. This is perhaps best epitomized by Bailey's "Twelve Ways to Obfuscate the Performance of a Parallel Machine" [12].

Nevertheless, benchmarks provide insight into the relative performance of different parallel computers. One of the most widely cited is the **NAS Parallel Benchmarks** [13, 14] which includes five kernels (two dimensional statistics from a Gaussian pseudo-random number generator, multigrid 3-D Poisson equation, conjugate gradient methods computation of the smallest eigenvalue of a large sparse symmetric positive definite matrix, 3-D Fast Fourier Transform, and integer sort) and three simulated CFD applications (SSOR algorithm for block 5 × 5 system, scalar pentadiagonal system, and block tridiagonal system). The NAS Parallel Benchmarks are described algorithmically, rather than in a specific programming language[3].They have been executed on numerous machines including Convex Exemplar SPP1000, Cray C90/T90/J90/T3D, DEC Alpha Server 8400, Fujitsu VPP500, IBM SP2 (Thin and Wide Node) and SGI Power Challenge XL. Saini and Bailey [16] make several observations. These include 1) the performance per unit cost (*e.g.*, MFlops per dollar) of the Cray C-90 was the *lowest* of all systems tested[4], and 2) all vendors employed their own specialized parallelization directives to achieve maximum performance. Future enhancements to the NAS Parallel Benchmarks include the development of a version using High Performance Fortran and Message Passing Interface (see below).

3.5 Portability

In recent years, significant effort has been devoted to the development of standardized environments for development of parallel codes. Three specific areas of activity are discussed here, namely, development of a standard Fortran for parallel computing (HPF), a standard for heterogeneous, network-based parallel computing environments (PVM), and the more recently developed standard message-passing interface MPI. There are many other similar research efforts in progress; however, space does not permit their discussion here.

[1] An alternate definition of efficiency (denoted as **scaled effiency**, and it counterpart, **scaled speedup**) has been proposed whereby the ratio of communications cost to computational cost remains fixed as n is increased. This is achieved by increasing the problem size (*i.e.*, N) with the number of processors. From the above analysis, this implies that $N^3 \sim n$.

[2] A compendium of benchmark reports is available at http://performance.netlib.org/performance/-html/PDSreports.html.

[3] In contrast, for example, to the LINPACK benchmark [15] for the matrix of order 100 which is written in FORTRAN and which may not be modified, including the comment statements.

[4] The system cost is assumed to be the list price.

3.5.1 High Performance Fortran

High Performance Fortran [17, 18, 19] is a data parallel language which extends Fortran 90 to provide additional support for the data parallel programming style while maintaining compatibility[5] with Fortran 90. Development of HPF was initiated in 1991 through the establishment of the High Performance Fortran Forum, and the language specification was published in May 1993. At present, twelve vendors have announced support of HPF. Additional information may be obtained at http://www.erc.msstate.edu/hpff/home.html. A good introduction to HPF is provided by Foster [21].

HPF extends Fortran 90 to include specific compiler directives to control the alignment and distribution of data on parallel machines, and introduces new parallel features and additional intrinsic library functions. For example, the `PROCESSORS` directive specifies the shape and size of an array of (abstract) processors, and the `ALIGN` directive aligns elements of different arrays with each other, thereby indicating that they should be distributed across processors in the same manner. New intrinsic functions introduced by HPF include `NUMBER_OF_PROCESSORS` and `PROCESSORS_SHAPE` which allow a program to obtain information on the number of processors on which it executes and the connection topology.

Examples of applications written in HPF are presented in Hawick and Fox [22] and Mueller and Ruehl [23]. A more extensive list is available on http://www.npac.syr.edu/hpfa/bibl.html.

3.5.2 Parallel Virtual Machine (PVM)

A recent major advancement is the development of heterogeneous, network-based parallel computing environments. Unlike fixed parallel computer architectures (*e.g.*, Cray C-90, Intel Paragon, etc.), these network-based parallel computers are created as a *virtual machine* using software tools such as PVM, Linda, P4 or Express. Typically, any number of different networked computers may be connected to form a parallel machine, although usually the computers are fairly similiar. Below, we provide a brief description of PVM. Description of other systems are available (*e.g.*, [24, 25, 26]), and a reasonably comprehensive listing has been complied by Turcotte [27]. Comparisons of the relative merits of different systems have also been published (*e.g.*, [28]).

PVM (Parallel Virtual Machine), created by the Heterogeneous Network Project (Oak Ridge National Laboratory, the University of Tennessee and Emory University) initiated in 1989, consists of two software packages [29, 30, 31, 32, 33]. The first is a daemon *pvmd3* which executes on all of the computers which comprise the virtual parallel machine. PVM is designed to enable any user with a valid login to install and initiate *pvmd3*. The user specifies a list of computers which comprise the virtual parallel machine, and starts *pvmd3* on each one. The PVM application can then be initiated from any of the computers. The second is a library of PVM routines *libpvm3.a* which contains the user callable routines for message pasing, spawning processes, coordinating tasks and modifying the virtual machine.

PVM has been successfully implemented on numerous computer architectures [33]. These include heterogeneous and homogeneous networks of computers, and also "individual" massively parallel computers (*e.g.*, Intel Paragon and Cray T3D). PVM is widely utilized in academia, industry and government laboratories. It is estimated that more than 10,000 individuals or installations have obtained the PVM software and approximately 20% to 25% are actively using it [34]. An index of PVM software may be obtained by sending the message `send index from pvm3` to `netlib@ornl.gov`.

An example of a PVM application is the Korringa, Kohn and Rostoker coherent potential approximation (KKR-CPA) method for computing the electronic properties, energetics and other ground state properties of substitutionally disordered alloys [33]. An approximate three month effort converted the 20K line KKR-CPA code for PVM. The code achieved approximately 200 MFlops using a network of ten IBM RS/6000 (6 model 530's and 4 model 320's) workstations, which is estimated to be approximately 82% of the maximum floating point capability of this virtual system. Also, the PVM KKR-

[5] For a description of Fortran 90, see Metcalf and Reid [20].

CPA code achieved over 9 GFlops performance using a network of twenty seven Cray C-90 and Cray Y-MP processors scattered across several sites. Furthermore, the PVM KKR-CPA code was successfully demonstrated for a virtual machine consisting of two Intel Paragons, a CM-5, an Intel i860 and IBM workstations, which were geographically distributed at several sites.

Load balancing, latency and bandwidth are clearly important issues for implementation of a virtual machine with PVM or other similar tools. In a heterogeneous environment, due consideration of the relative performance of individual hosts is obviously needed in domain decomposition. Latency (*i.e.*, the time required to initiate a message) can be a critical issue, depending on the ratio of communications to computation. Network bandwith may be restricted due to existing traffic. Recent enhancements to PVM [34] provide for improved performance. For example, the message passing performance of PVM on the Intel Paragon[6] is only 5% to 8% slower than the native functions [34].

3.5.3 Message Passing Interface (MPI)

MPI (Message Passing Interface) is a message passing standard for homogeneous and heterogeneous parallel and distributed computing systems. The development of the MPI standard is a multinational effort which was initiated in 1992 and is supported by ARPA, NSF and the Commission of the European Community. The MPI standard was published in 1994 and is described in [35, 36, 37]. A good introduction to MPI is provided by Foster [21], and a brief description is presented in [38].

An MPI program includes one or more processes which communicate with each other through calls to MPI library routines. There are two types of communications, namely, point-to-point communication between pairs of processes, and collective communication between groups of processes. Several variants of "send" functions are provided to enable users to achieve peak performance. Two basic types of communications topologies are provided: a cartesian grid and an arbitrary process graph [38].

Due to its recent introduction, there are a relatively small number of applications to date using MPI. A recent review by Skjellum, Lusk and Gropp [39] describes recent applications including unsteady incompressible viscous flows, groundwater modeling, volume visualization and traffic simulation. Native MPI implementations are currently under development by several parallel computer vendors [40].

4 Parallel Computing in Aerospace Research

Despite the extensive research on parallel computing, only a small fraction of numerical simulations of aerospace *research* problems employ parallel computing. A survey of the citations for parallel and other computers for three journals is presented in Table 3. The period July 1993 through July 1995 was surveyed for all articles presenting research involving significant numerical simulation. Approximately 44% of these articles indicated that a serial or vector machine (single processor) was employed, while only 3.4% specifically noted that a parallel computer was used. Approximately 52% did not indicate that machine used. If the statistics for the first two categories are assumed statistically representative of the last group, than an overall estimate (upper bound) for the parallel applications is 7%.

Why are so few research simulations performed on parallel computers ? Certainly, research *on* parallel computing has shown the capability for solving a wide range of fluid dynamics problems. At the Parallel CFD '95 Conference, applications of parallel computing were presented for reacting flows, Euler and Navier-Stokes solvers, spectral methods, multigrid methods, and adaptive schemes. Numerous other applications have been developed (*e.g.*, see, for example, [41, 42, 43, 44]).

I posed this question to a number of experts in parallel CFD. The answers tended to be fairly similar, and not at all surprising. All focused on the issue of *calendar time required to solve a particular problem.* As one person stated, "The machine which you use to solve a problem is irrelevant. The only thing that matters is how quickly you can get the problem done." At the

[6] Using the functions `pvm_psend()` and `pvm_precv()` introduced in PVM Version 3.3.

Table 3: Citations of Parallel and Other Computers (July 93 - July 95)

Journal	*Parallel*	*Serial/Vector*	*Not Stated*	*Total*
AIAA Journal	7	104	110	221
Journal of Aircraft	0	48	47	95
Journal of Fluid Mechanics	8	44	76	128
Total	15	196	233	444
Percent	3.4%	44.1 %	52.5%	100.0 %

present time, many CFD researchers who are not using parallel computing view parallel CFD as 1) lacking a decisive advantage performance advantage (*e.g.*, MFlops) over conventional serial (and vector) computers in many instances, 2) difficult to program efficiently, and 3) lacking in portability.

All of these factors are likely to diminish in the near future, and thus the use of parallel computing in aerospace research should increase. Microprocessor CPU performance continues to improve by a factor of 1.5 to 2.0 per year[7] [45], and consequently parallel machines are now comparable or faster than traditional vector supercomputers. For example, the Cray T3D (512 processors) is on average 41% faster[8] than the Cray C-90 (16 processors) for the three simulated CFD applications in the NAS Parallel Benchmarks. The Cray T-3D (1024 processor) is 128% faster. The IBM SP2-WN (160 processors) was also significantly faster than the Cray C-90 (16 processors) [16]. Also, the emergence of standards in parallel programming languages (*e.g.*, HPF) and message passing functions (*e.g.*, PVM, MPI) simplify the development of parallel code and significantly enhance its portability.

5 Parallel Computing in Aerospace Industry

Parallel computing has a major presence in the aerospace industry. Within the past few years, several major aerospace corporations have developed extensive **Networks of Workstations (NOWs)** for production analysis and design. Two examples are Pratt & Whitney (East Hartford, CT, and Palm Beach, FL) and McDonnell Douglas Aerospace (St. Louis, MO).

Pratt & Whitney (P&W) initiated their Network of Workstations concept [46] in mid-1989. The decision was motivated by two factors. First, P&W had an installed base of workstations which had been acquired principally for design/drafting work, but which were effectively unused in the evenings and on weekends. Thus, there was a surplus of compute cycles which could be employed for analysis and design, provided that the computational tasks could be decomposed and parallelized. Second, their existing Cray X-MP, purchased in 1986, was both severely overloaded and limited in capability (*e.g.*, memory). Hence, there was a significant incentive to invest resources in development of a new paradigm for CFD analysis and design.

The P&W approach consists of several parts. The flow solver is **NASTAR**, a 3-D structured grid multi-block Navier-Stokes code. Domain decomposition is straightforward, *i.e.*, each block is assigned to an individual processor (workstation). An example is shown in Fig. 3. The momentum, energy and turbulence scalar equations are solved using Successive Line Under Relaxation (SLUR). The SLUR iterations are performed independently within each block, with periodic updating of the boundary conditions to transmit information between blocks. The optimal updating strategy is found by numerical experiments. The pressure correction equation is solved to satisfy the continuity equation, and employs a parallelized Preconditioned Conjugate Residual (PCR) algorithm. The majority of the computational effort is expended in the pressure correction equation, and thus considerable effort was focused on efficient paral-

[7] The rate of improvement of microprocessor performance is much faster than for the specialized processors developed for traditional vector machines (*e.g.*, Cray C-90)

[8] *I.e.*, the ratio of the execution time on the Cray C-90 to the Cray T3D was 1.41.

lelization of the PCR algorithm. Management of the individual block computations is performed by **Prowess** (Parallel Running of Workstations Employing SocketS), developed by P&W, which provides communications, parallel job process control, accounting, reliability and workstation user protection. Communications between individual workstations is performed directly using sockets which emulate a file I/O paradigm. Checkpointing is employed to achieve high reliability. Workstation user protection is the implementation of the P&W policy that the interactive user has the first priority on a workstation. Thus, for example, Prowess suspends (or terminates) any remote process executing on a workstation as soon as any activity is detected on the workstation's keyboard or mouse. Idle worksations capable of executing NASTAR are identified using the **Load Sharing Facility (LSF)** software from Platform Computing Corporation.

The P&W workstation network employed for parallel computing is substantial. Approximately 400 to 600 workstations are employed daily for parallel CFD jobs at P&W's East Hartford, CT facility, and another 300 to 400 workstations at Palm Beach, FL. The growth in usage of the workstation network for parallel CFD application is displayed in Fig. 7.

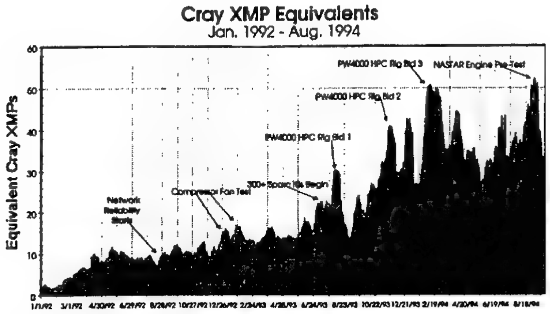


Figure 7: Daily parallel CFD throughput on Pratt & Whitney's East Hartford, CT workstation network (from [46])

A critical element in the Network of Workstations approach is the network configuration. Adequate communications bandwith is essential for effective distributed parallel computing. The P&W East Hartford, CT network architecture is shown in Fig. 8. It includes multiple Fiber Distributed Data Interface (FDDI) 100 Mbps backbone networks connected by Digitial Equipment FDDI Gigaswitches. There are approximately 200 ethernet segments.

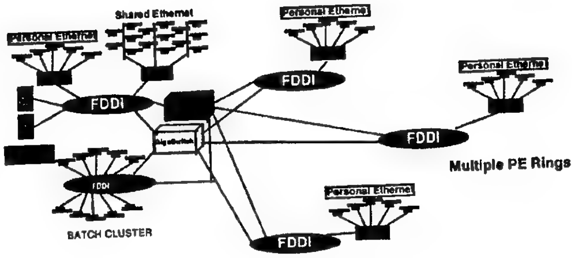


Figure 8: Pratt & Whitney network backbone in East Hartford, CT (from [46])

Pratt & Whitney has concluded that their Network of Workstations paradigm has been successful. Fischberg *et al* [46] cite a reduction in design time of 50% to 67% for a high pressure compressor and fan design, respectively.

McDonnell Douglas Aerospace initiated their Network of Workstations concept [47] in late 1992. The decision was motivated by factors similar to Pratt & Whitney. First, McDonnell Douglas had a substantial number of workstations (mostly Hewlett-Packard 7xx, plus a small number of IBM RS/6000 and Silicon Graphics) which had been acquired principally for CAD. These workstations were typically utilized during the daytime, and largely unused in evenings and on weekends. Second, their existing Cray X-MP/18 was both heavily loaded and limited in capability (*e.g.*, memory), and the corporate financial position precluded a multi-million dollar new supercomputer acquisition. Third, McDonnell Douglas wanted to gain experience with parallel computing technology.

The McDonnell Douglas Aerospace approach consists of several parts. The flow solver is **NASTD**, a proprietary 3-D structured grid multi-block compressible Euler/Navier-Stokes code. The code is heavily utilized at McDonnell Douglas, with typically fifty active users. A straightforward domain decomposition is employed, whereby a grid block (subdomain) is assigned to an individual processor (workstation). The code is operated in a master/slave relationship using PVM [32] for process control and explicit message passing between processors.

Parallel computations using NASTD are performed in the evenings and on weekends using up to 400 workstations in clusters of 15 to 20 workstations per job. The reliability (*i.e.*, the percentage of submitted jobs which complete successfully) exceeds 95%. Numerous difficulties were resolved in achieving this performance, many of which were management issues, *e.g.*, negotiating scheduled hardware, software and network maintenance (which had oftentimes occurred at random intervals at nights and on weekends), and changing the perception that the individual user "owned" the workstation and could therefore reboot it whenever desired (thus terminating any slave process in operation and crashing the entire parallel computation).

6 Conclusions

Several main conclusions can be drawn regarding parallel computing in CFD:

- There is a large number of vendors of parallel computers whose systems offer a wide range of performance.
- Modern parallel computers can equal or exceed the performance of the largest multiprocessor Cray supercomputers.
- The aerospace industry has taken a leading role in the application of parallel computing to practical analysis and design.
- The aerospace research community (*e.g.*, academia and research laboratories) has taken a leading role in research *on* parallel computing, but has not significantly employed parallel computing in solving aerospace research problems.
- The development of message passing standards (*e.g.*, PVM and MPI) and data parallel programming language standards (*e.g.*, HPF) will expand use of parallel computing in CFD.

7 Acknowledgments

I am indebted to R. Agarwal, T. Barth, R. Cosner, C. Fischberg, J. Lewis, R. Pelz, J. Shang, H. Simon, and V. Venkatakrishnan for their comments and suggestions, and to H. Lau for the journal survey.

References

[1] R. Hockney and C. Jesshope, *Parallel Computers 2*. Bristol, England: Adam Hilger, 1988.

[2] A. W. Burks and A. R. Burks, "The ENIAC: the First General-Purpose Electronic Computer," *Ann. Hist. Comput.*, vol. 3, pp. 310–399, 1981.

[3] M. Flynn, "Some Computer Organizations and Their Effectiveness," *IEEE Transactions on Computers*, vol. C-21, pp. 948–960, 1972.

[4] G. Bell, "Ultracomputers: A Teraflop Before Its Time," *Communications of the ACM*, vol. 35, pp. 27–47, August 1992.

[5] J. Shore, "Second Thoughts on Parallel Processing," *Comput. Elect. Engr.*, vol. 1, pp. 95–109, 1973.

[6] S. Borrelli, A. Matrone, and P. Schiano, "A Multiblock Hypersonic Flow Solver for Massively Parallel Computer," in *Parallel Computational Fluid Dynamics '92* (R. Pelz, A. Ecer, and J. Hauser, eds.), (New York), pp. 25–37, Elsevier Science B. V., 1993.

[7] P. Schiano and A. Matrone, "Parallel CFD Applications: Experiences on Scalable Distributed Multicomputers," in *Parallel Computational Fluid Dynamics: New Trends and Advances* (A. Ecer, J. Hauser, P. Leca, and J. Periaux, eds.), (New York), pp. 3–12, Elsevier Science B. V., 1995.

[8] H. Simon, W. V. Dalsem, and L. Dagum, "Parallel CFD: Status and Future Requirements," in *Parallel Computational Fluid Dynamics: Implementations and Results* (H. Simon, ed.), pp. 1–29, MIT Press, 1992.

[9] V.Venkatakrishnan, H. Simon, and T. Barth, "A MIMD Implementation of a Parallel Euler Solver for Unstructured Grids." Technical Report RNR-91-24, NASA Ames Research Center, 1991.

[10] J. McDonald, "Particle Simulations in a Multiprocessor Environment." Technical Report RNR-91-02, NASA Ames Research Center, 1991.

[11] G. Amdahl, "The Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities," in *AFIPS Conference Proceedings Spring Joint Computing Conference*, vol. 30, pp. 483–485, 1967.

[12] D. Bailey, "Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers," *Supercomputing Review*, pp. 54–55, August 1991.

[13] D. B. *et al*, "The NAS Parallel Benchmarks," *International Journal of Supercomputer Applications*, vol. 5, no. 3, pp. 63–73, 1991.

[14] D. Bailey, J. Barton, T. Lasinski, and H. Simon, "The NAS Parallel Benchmarks," Tech. Rep. NASA TM 103863, NASA Ames Research Center, July 1993.

[15] J. Dongarra, "Performance of Various Computers Using Standard Linear Equations Software." Report CS-89-95, Computer Science Department, University of Tennessee, and Oak Ridge National Laboratory, 1995.

[16] S. Saini and D. Bailey, "NAS Parallel Bemchmarks Results 3-95," Tech. Rep. NAS 95-011, NASA Ames Research Center, April 1995.

[17] H. P. F. Forum, "High Performance Fortran Language Specification," *Scientific Programming*, vol. 2, pp. 1–170, 1993.

[18] C. Koelbel, D. Loveman, R. Schreiber, G. Steele, and M. Zosel, *The High Performance Fortran Handbook.* Cambridge, MA: The MIT Press, 1994.

[19] H. Performance Fortran Forum, "High Performance Fortran Language Specification," tech. rep., Center for Research on Parallel Computing, Rice University, November 1994.

[20] M. Metcalf and J. Reid, *Fortran 90 Explained.* New York: Oxford Science Publications, 1990.

[21] I. Foster, "Designing and Building Parallel Programs." http://www.mcs.anl.gov/-dbpp/.

[22] K. Hawick and G. Fox, "Exploiting High Performance Fortran for Computational Fluid Dynamics," Tech. Rep. SCCS-661, Northeast Parallel Architecture Center, November 1994.

[23] A. Mueller and R. Ruehl, "Extending High Performance Fortran for the Support of Unstructured Computations," in *Proc. of 9th ACM Inter. Conf. on Supercomputing*, 1995.

[24] N. Carriero and D. Gelertner, "Linda in Context," *Communications of the ACM*, vol. 32, pp. 444–458, April 1989.

[25] R. Butler and E. Lusk, "User's Guide to the P4 Programming System." Argonne National Laboratory, Technical Report ANL-92/17, 1992.

[26] A. Kolawa, "The Express Programming Environment." Workshop on Heterogeneous Network-Based Concurrent Computing, Tallahassee, October 1992.

[27] L. Turcotte, "A Survey of Software Environments for Exploiting Networked Computing Resources." Engineering Research Center for Computational Field Simulations, Mississippi State University, January 1993.

[28] T. Mattson, "Programming Environments for Parallel and Distributed Computing: A Comparison of p4, PVM, Linda and TCGMSG," *The International Journal of Supercomputing*, vol. 9, pp. 138–161, September 1995.

[29] V. Sunderam, G. Geist, J. Dongarra, and R. Manchek, "PVM: A Framework for Parallel Distributed Computing," *Journal of Concurrency: Practice and Experience*, vol. 2, pp. 315–339, December 1990.

[30] G. Geist and V. Sunderam, "Network Based Concurrent Computing on the PVM System," *Journal of Concurrency: Practice and Experience*, vol. 4, pp. 293–311, June 1992.

[31] J. Dongarra, G. Geist, R. Manchek, and V. Sunderam, "Integrated PVM

Computing," *Computers in Physics*, vol. 7, no. 2, pp. 166–175, 1993.

[32] A. G. *et al*, "PVM3 User's Guide and Reference Manual." Oak Ridge National Laboratory, 1994.

[33] V. Sunderam, G. Geist, J. Dongarra, and R. Manchek, "The PVM Concurrent Computing System: Evolution, Experiences and Trends," *Journal of Parallel Computing*, vol. 20, no. 4, pp. 531–547, 1994.

[34] A. Beguelin, J. Dongarra, A. Geist, R. Manchek, and V. Sunderam, "Recent Enhancements to PVM," *The International Journal of Supercomputing*, vol. 9, pp. 108–127, September 1995.

[35] M. P. I. Forum, "MPI: A Message-Passing Interface Standard," *Journal of Supercomputer Applications*, vol. 8, 1994.

[36] M. P. I. Forum, "MPI: A Message-Passing Interface Standard," tech. rep., University of Tennessee, 1994.

[37] W. Gropp and E. Lusk, "Message Passing Interface." http://www.mcs.anl.gov/mpi, 1994.

[38] L. Clarke, I. Glendinning, and R. Hempel, "The MPI Message Passing Interface Standard," March 1994. ftp://par.soton.ac.-uk/pub/mpi/paper.ps.

[39] A. Skjellum, E. Lusk, and W. Gropp, "Early Applications in the Message-Passing Interface (MPI)," *The International Journal of Supercomputer Applications*, vol. 9, no. 2, pp. 79–94, 1995.

[40] W. Gropp and E. Lusk, "Implementing MPI: the 1994 MPI Implementors' Workshop." http://www.mcs.anl.gov/mpi-impl/paper.ps, October 1994.

[41] H. Simon, ed., *Parallel Computational Fluid Dynamics*, MIT Press, 1992.

[42] R. Pelz, A. Ecer, and J. Hauser, eds., *Parallel Computational Fluid Dynamics '92*, Elsevier N.H., 1993.

[43] A. Ecer, J. Hauser, P. Leca, and J. Periaux, eds., *Parallel Computational Fluid Dynamics*, (New York), Elsevier N.H., 1995.

[44] D. B. *et al*, ed., *Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing*, SIAM, 1995.

[45] J. Hennessy and N. Jouppi, "Computer Technology and Architecture: an Evolving Interaction," *IEEE Computer*, vol. 24, no. 9, pp. 18–29, 1991.

[46] C. Fischberg, C. Rhie, R. Zacharias, P. Bradley, and T. DesSureault, "Using Hundreds of Workstations for Production Running of Parallel CFD Applications," in *Parallel Computing 1995*, 1995.

[47] R. Cosner, "Experiences at McDonnell Douglas in Converting CFD Production to Parallel Processing," in *Parallel Computing 1995*, 1995.

Portable Parallelization of a 3D Euler/Navier-Stokes Solver for Complex Flows

B. Eisfeld, H. Ritzdorf*, H. Bleecke, N. Kroll

Institute of Design Aerodynamics
DLR, D-38108 Braunschweig
Germany

*Institute SCAI
GMD St. Augustin, Germany

SUMMARY

This paper describes the portable parallelization of the FLOWer code, a large, block structured CFD solver for industrial use. Basic requirements for the parallelization are identified, and the strategies applied for its parallelization are explained. Special emphasis is put on the parallel heart of the program, the communications library CLIC-3D. Results obtained on several platforms demonstrate the success of the method chosen and allow an assessment of today's capabilities of parallel computers in CFD applications. Parallel computations of aircraft configurations of varying complexity prove that parallel computers have become operational in aircraft development.

LIST OF SYMBOLS

C_p	specific heat at constant pressure
$\vec{D}$	vector of artificial dissipative fluxes
E	total energy
$\bar{\bar{F}}$	flux tensor
H	total enthalpy
k	heat transfer coefficient
N_B	number of blocks
$\vec{n}$	outward pointing unit normal vector
Pr	Prandtl number
p	pressure
$\vec{q}$	velocity vector
$\vec{R}$	residual vector
S	speed-up
T	temperature
t	execution time
u	velocity in x-direction
V	volume
v	velocity in y-direction
$\vec{W}$	vector of conservative variables
w	velocity in z-direction
γ	ratio of specific heats
μ	viscosity
ρ	density
σ	normal stress components
τ	shear stress components
ϕ	components of the energy dissipation function

Indices

alg	algorithmic ideal
ijk	discrete point
l	laminar
t	turbulent
x	in x-direction
y	in y-direction
z	in z-direction
∞	at infinity

1. INTRODUCTION

When looking on the progress made in CFD during the last decade, one observes that improvements are made in two directions: The algorithms became more flexible and faster, e. g. by multigrid techniques, and the hardware platforms increased in main memory and CPU performance. As far as the progress in computer power is concerned, experts predict that only parallel architectures will allow further improvements leading to peak performances of about 1 TFLOP/s [1, 2].

Therefore, since this type of super computers might require a new type of application software, the development of parallel flow solvers is mandatory, if one wants to exploit their abilities in the future. This could be treated as an isolated subject, when dealing with questions of basic research interest, but when concerning large codes in industrial use, several constraints are limiting the development.

First of all the effort spent for parallelization must be justified by the gain in compute power or the reduction of computing costs, respectively. Secondly, large CFD

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

solvers usually have been developed throughout a long period involving a number of different scientists, and they are applied by numerous users which both must be respected by a parallelization. Last but not least, there is not just one parallel architecture available at the moment, but the platforms differ in the design of the CPUs (vector versus RISC processors), the memory organization (shared versus distributed memory) and the communication systems (hardware and software). Therefore, if one wants to be able to follow any hardware development in the future, one must keep the parallelization as flexible as possible.

The paper presented here describes the portable parallelization of the FLOWer code which is currently carried out within the project POPINDA (POrtable Parallelization of INDustrial Aerodynamical applications) funded by the German Ministry of Research (BMBF). The FLOWer code is a block structured CFD solver for complex flows in configuration aerodynamics. It has directly evolved from the DLR-CEVCATS code [3] and is developed in close cooperation of the DLR with the German national research center for computer science (GMD) and the German aeronautical industry (DASA) as a multi purpose flow solver.

After a description of the numerical algorithm of this large CFD code in the next section, the strategy chosen for its parallelization will be explained outlining the ideas of how to meet the requirements for large application programs in industrial use. The communications library CLIC-3D which solves the portable parallelization problem of such codes is then reviewed.

Benchmark results obtained on various platforms demonstrate the portability of the FLOWer code and allow an assessment of today's parallel platforms. Furthermore, computations of different aircraft configurations show that such architectures have become operational for CFD applications and what effects on the obtainable performance occur. Finally, the computation of a 6 million grid point test case on 129 processors indicate the future potential of parallel processing in CFD.

2. NUMERICAL METHOD OF THE FLOWer CODE

2.1 Governing Equations

The FLOWer code is solving the Euler- or Navier-Stokes equations in conservative form [3, 4] written as

$$\frac{\partial}{\partial t}\int_V \vec{W}\,dV + \int_{\partial V} \bar{\bar{F}} \cdot \vec{n}\,dS = 0 \qquad (1)$$

where $\vec{W}$ denotes the vector of conservative variables

$$\vec{W} = \begin{pmatrix} \rho & \rho u & \rho v & \rho w & \rho E \end{pmatrix}^T \qquad (2)$$

and $\bar{\bar{F}}$ is the flux tensor being defined by

$$\bar{\bar{F}} = \begin{pmatrix} \rho u & \rho v & \rho w \\ \rho u^2 + p + \sigma_x & \rho uv & \rho uw \\ \rho uv & \rho v^2 + p + \sigma_y & \rho vw \\ \rho uw & \rho vw & \rho w^2 + p + \sigma_z \\ \rho uH + \phi_x & \rho vH + \phi_y & \rho wH + \phi_z \end{pmatrix} \qquad (3)$$

with the abbreviations

$$\begin{aligned} \phi_x &= u\sigma_x + v\tau_{xy} + w\tau_{xz} - k\frac{\partial T}{\partial x} \\ \phi_y &= u\sigma\tau_{xy} + v\sigma_y + w\tau_{yz} - k\frac{\partial T}{\partial y} \\ \phi_z &= u\tau_{xz} + v\tau_{yz} + w\sigma_z - k\frac{\partial T}{\partial x} \end{aligned} \qquad (4)$$

The elements of the viscous stress tensor are determined by Newton's law of skin friction, i. e.

$$\begin{aligned} \sigma_x &= -2\mu\frac{\partial u}{\partial x} + \frac{2}{3}\mu\vec{\nabla}\cdot\vec{u} \\ \sigma_y &= -2\mu\frac{\partial v}{\partial y} + \frac{2}{3}\mu\vec{\nabla}\cdot\vec{u} \\ \sigma_z &= -2\mu\frac{\partial w}{\partial z} + \frac{2}{3}\mu\vec{\nabla}\cdot\vec{u} \\ \tau_{xy} &= -\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \tau_{yz} &= -\mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\ \tau_{xz} &= -\mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right) \end{aligned} \qquad (5)$$

Further simplification is obtained by applying a thin shear layer approximation accounting only for gradients normal to surfaces.

For the non-dimensional pressure and temperature the following relations hold

$$\begin{aligned} p &= \rho\,(\gamma - 1)\left(E - \frac{\vec{q}^{\,2}}{2}\right) \\ T &= \frac{p}{\rho} \end{aligned} \qquad (6)$$

and the system is closed by the relations for the transport coefficients

$$\mu = \mu_l + \mu_t$$
$$k = C_p\left(\frac{\mu_l}{Pr_l} + \frac{\mu_t}{Pr_t}\right) \qquad (7)$$

where the laminar viscosity μ_l is given by Sutherlands's formula

$$\frac{\mu_l}{\mu_\infty} = \left(\frac{T}{T_\infty}\right)^{3/2} \frac{T_\infty + 110K}{T + 110K} \qquad (8)$$

In turbulent flows the eddy viscosity μ_t is computed from the algebraic Baldwin-Lomax model [5].

2.2 Discretization and Time Integration

The governing equations are discretized by the method of lines separating the space and time coordinates. After the space discretization, a system of ordinary differential equations in time results involving each finite volume. For any hexaeder of the structured grid one obtains the equation.

$$\frac{d}{dt}\vec{W}_{ijk} + \frac{1}{V_{ijk}}\int_{\partial V} \bar{\bar{F}}_{ijk} \cdot \vec{n}dS = 0 \qquad (9)$$

The space discretization is central, so that an artificial dissipation term due to Jameson et al. [6] is added damping high frequency oscillations and allowing a sufficiently sharp resolution of shock waves in the flow field. The resulting system of equations then reads

$$\frac{d}{dt}\vec{W}_{ijk} + \frac{1}{V_{ijk}}\left(\vec{R}_{ijk} - \vec{D}_{ijk}\right) = 0 \qquad (10)$$

with $\vec{R}_{ijk}$ and $\vec{D}_{ijk}$ being the vector of the residuals and the artificial dissipative fluxes respectively.

The time integration is carried out by an explicit, hybrid multi stage Runge-Kutta scheme which is accelerated by the techniques of local time stepping, enthalpy damping (Euler) and implicit residual smoothing [7].

This procedure is embedded into a powerful multigrid algorithm [3, 8] which allows standard single grid computations as well as a successive grid refinement and simple or full multigrid, respectively. As is illustrated in [3], where a more detailed description can be found, high convergence rates can be obtained, using this technique.

2.3 Block Structure

Since structured grids around complex geometries cannot be generated as one logically rectangular domain, the FLOWer code is block structured. That means that the flow field is split into regions for each of which the generation of a structured grid is possible. Figure 1 is showing schematically such a grid topology around a transport aircraft. As one can see, the flow field is subdivided into four areas of similar size around the wing body. Three subdomains are covered by one block each (blocks 1 to 3), whereas the fourth region is further subdivided, due to the presence of an engine there (blocks 4 to 9). The engine is surrounded by a polar grid (blocks 8 and 9) which is adapted by blocks 6 and 7 to the general O-H topology (blocks 3 to 5).

The program then treats the blocks more or less independently of each other which can only be done properly by exchanging data of the current solution at block interfaces before each time step.

Therefore, the blocks are surrounded by layers of dummy cells, which at block intersections correspond with the physical cells of the neighboring domain.The FLOWer code allows an overlap width of two cells resulting in second order accuracy of the scheme at those boundaries. This is necessary, in order to treat the artificial dissipation terms correctly which otherwise could spoil the solution as shown in [9].

Currently, the FLOWer code allows different exchange strategies for the data at block intersections varying in effort and accuracy [10].

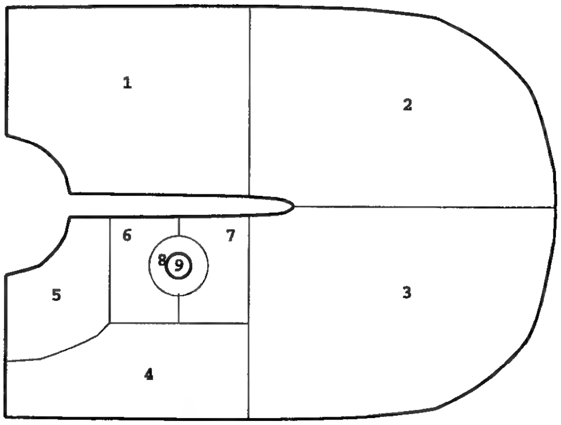


Fig. 1 Schematic multiblock decomposition of the flow field around a generic transport aircraft. Decomposition into 9 blocks due to the adaption of an engine fitted polar mesh to a global O-H topology.

3. PARALLELIZATION OF THE FLOWer CODE

3.1 Requirements

When parallelizing a large CFD solver as the FLOWer code, the parallelization cannot be treated in isolation, but must be integrated into the general development procedure [9]. Therefore, certain objectives must be met, the most important of which are specified in the following.

Portability

The FLOWer code is developed by a number of scientists working at different locations on a variety of computers. Furthermore, it is applied by several users running the program on other platforms than the developers. Finally, the life time of the program will certainly exceed that one of most of today's computers, so that portability is a major requirement:

The FLOWer code must run on any platform, it may be sequential or parallel !

Conservation of the development history

When developing the parallel FLOWer code, its algorithm had already reached a high degree of maturity established by various scientists during a long period within the DLR-CEVCATS code. Moreover, the users had become experienced with its handling and in interpreting its results. Therefore, the parallelization had to respect that development history:

The FLOWer code must not be completely re-written due to its parallelization !

Low parallelization effort

Parallelization is only one means of high performance computing and should not be done just for its own sake. The effort spent for parallelization must therefore be justified by the corresponding gain in performance or reduction in computational costs, respectively:

The parallelization of the FLOWer code must achieve the highest performance possible at lowest costs !

3.2 Parallelization Strategies

Parallelization of a CFD solver means mapping of inherent parallelism incorporated in the program to a parallel architecture using a communication model. As far as structured codes are concerned, there is parallelism on statement level (multiply / add), in the data (loops over all points of a block) and in the geometry (the different blocks) which can be expressed by parallelizing languages, e. g. HPF or C++, parallelizing compilers (directives / autotasking) or by message passing, i. e. by explicitly sending and receiving data to and from different processes. Moreover, the parallel hardware design varies with respect to the arrangement of CPUs, memory and the interconnecting network (shared / distributed memory, hybrid constructions) [9].

Therefore, one has to decide which type of parallelism should be exploited using which communication model, and how to achieve portability. When parallelizing the FLOWer code, general considerations led to the following guidelines allowing to meet the requirements stated above:

Grid partitioning as parallelization strategy

The idea is to map the different blocks to different processes where they are solved separately. Between the iteration steps the boundary data are exchanged via the network.

This technique is not only said to be efficient when solving partial differential equations [11, 12], but moreover guarantees the conservation of the sequential development history, because it is directly based on the sequentially well established multi block method.

Separation of computation and communication

A strict application of this rule allows an algorithmic development which remains independent from the parallelization or other hardware aspects. Additionally, the code structure can be kept modular more easily which is highly desired from software engineering reasons. Finally, the portability problem becomes much easier to handle, when concentrating the communication parts within separate units.

Communication by message passing

Besides efficiency arguments, the decision for the message passing communication model results mainly from the portability demand. Using a parallelizing language would have caused a complete re-implementation of the FLOWer code which was clearly unacceptable, and parallelizing compilers are only available on some platforms restricting the portability of the code.

In the contrary, it should be noted that the message passing approach does not exclude a parallelization by autotasking supported by compiler directives [9].

Use of a portable communications library

Combining the requirements for the parallelization of the FLOWer code with the above guidelines leads to the demand for a portable communications library

Such a high level library should perform all typical operations necessary in parallel mode involving communication between the different processes. Its usage should therefore guarantee parallel portability, keep the sequential code almost unchanged and reduce the parallelization effort drastically. Moreover, it should lead to a highly reliable parallelization.

This library has been developed by the GMD as CLIC-3D and will be described in the following section.

4. THE CLIC-3D COMMUNICATIONS LIBRARY

4.1 Background

The communications library CLIC-3D (Communications Library for Industrial Codes in 3 Dimensions) is currently developed by the GMD within the German research project POPINDA. It is based on the former GMD Comlib library and supports general block structured PDE solvers, particularly involving multigrid algorithms. Its development was based on the observation that for this class of programs the communication patterns are generally quite similar, although the numerical algorithms might differ considerably.

The major aim of the CLIC development is, to make programming for complex geometries as easy as for simple single block domains providing high level routines for all communication and mapping tasks. The CLIC user interface, therefore, provides the application program with all necessary data on the problem to be solved.

Currently, the CLIC library supports cell vertex and cell centered discretizations.

The portability of the CLIC library is achieved using the PARMACS as portable message passing interface [13]. This system was chosen, because it is a commercial product and not public domain as PVM [14], and MPI [15] was not yet available at the time, when the POPINDA project started. The corresponding software layers of the parallelized FLOWer code are illustrated in figure 2.

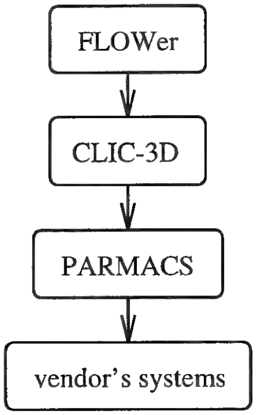


Fig. 2 Software layers of the parallel FLOWer code

4.2 General Code Structure

Since the CLIC library is based on the PARMACS message passing system, it is designed for a host-node (master-slave) programming model. The host process starts the distributed application on several nodes, performs the input and output and transfers data to and from the node processes. The host itself does not participate in the solution process which is exclusively carried out by the node processes. Consequently the user application program is seperated into a host and a node program as shown in figure 3.

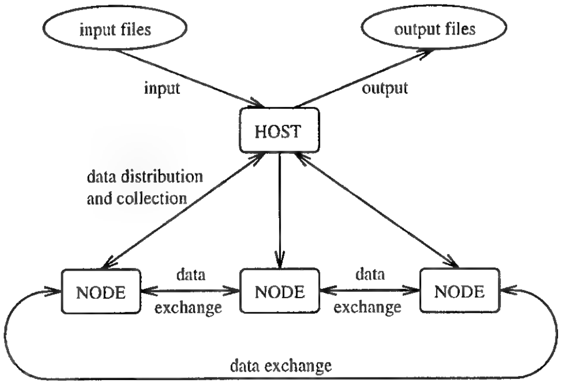


Fig. 3 Host-node structure of the parallel FLOWer code

The host program reads in the same input parameters as the sequential user program. Then, CLIC routines read in the description of the block structured grid, create the node processes and map the blocks onto the allocated node processors respecting load balance aspects as far as possible. Then, the input parameters are distributed to the node processes. Finally, another routine reads in the grid coordinates and sends them to the corresponding node processes only. After reading and distributing all data to the nodes, the host process waits for output generated by the node processes and writes it to the desired units.

Each node process executes an identical node program which may contain the complete sequential code. In case of the FLOWer code, the only differences in parallel mode are:

- The input data is not read in but received from the host process
- Global operations involving all blocks are passed to the CLIC library for performation
- The data exchange at block boundaries is carried out fully automatically by the CLIC library
- Write statements are replaced by parallel output routines of the CLIC library

A schematic flow chart of the parallel FLOWer code is given in figure 4.

Further activities of the CLIC library consist in the analysis of the given block structure, in order to allow a special treatment of grid singularities. For each segment edge and point the adjoining blocks and the number of adjoining cells is determined leading to a topological classification. If the segment is part of the physical boundary, the boundary conditions of all adjoining blocks are determined, additionally. Finally, geometrical singularities are detected, so that the user can inquire all data for a special treatment of irregular grid points.

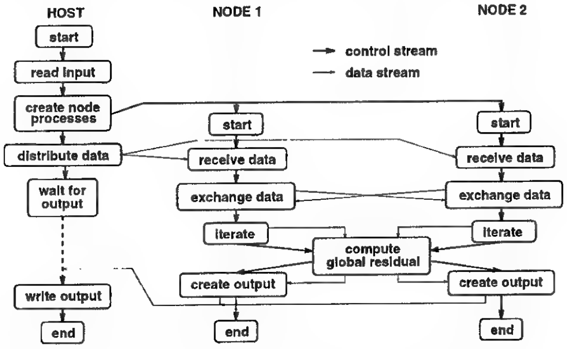


Fig. 4 Schematic flow chart of the parallel FLOWer code based on the CLIC library

The same data is needed by the CLIC library for optimization of the data exchange at block boundaries. The aim is, to send the minimum number of messages necessary for a correct update of the boundaries. This is important especially on coarse grids of multigrid algorithms where the communication may become significantly time consuming. Basic idea is the introduction of a global orientation for larger portions of the block structure leading to a fast exchange procedure. Only in topologically more complicated situations additional messages must be sent.

Another specialty of the CLIC library is the possibility of parallel output, i. e. output files can be directly written by the node processes.

4.3 Examples of High Level CLIC Operations

Exchange of boundary data

As already mentioned, the grid partitioning strategy requires an exchange of boundary data at the interfaces of the blocks. Therefore, the information on the topology of the block structure is stored in terms of block surface segments in a file that is read in by the CLIC library. During the initialization phase, this information is analyzed with respect to the necessary send and receive operations within a data exchange procedure.

When the corresponding exchange routine is called on each process, as sketched in figure 5, all interface data of the blocks on a process is stored segmentwise in a respective buffer which is sent (asynchroneously blocking) to the corresponding neighboring block on another process. Afterwards, messages of the other processes are received in the order they come in, and the buffers are unpacked. If necessary, the procedure is repeated for segment edges and corner points, so that finally all block interfaces are updated correctly.

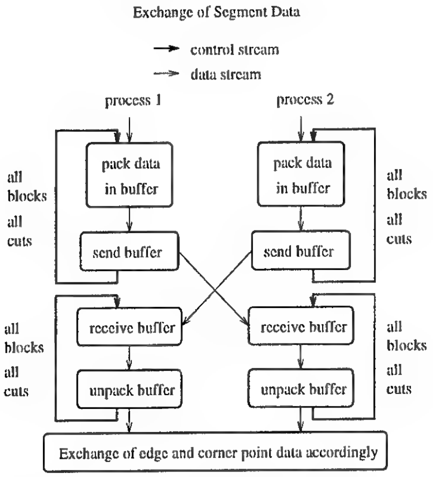


Fig. 5 Schematic flow chart for a parallel data exchange.

Global operation

Global operations involving all blocks of a given block structure are necessary, e. g. for the computation of a global residual. They are carried out within another special CLIC routine using an embedded binary tree for the process topology. As illustrated by figure 6, each parent process receives data from its child processes, performs a local operation on this data and communicates it to its own parent process. Afterwards, the process waits for a message from its parent process containing the correct global value which is obtained at the end of the chain. After its reception, this value is further communicated to the corresponding child processes.

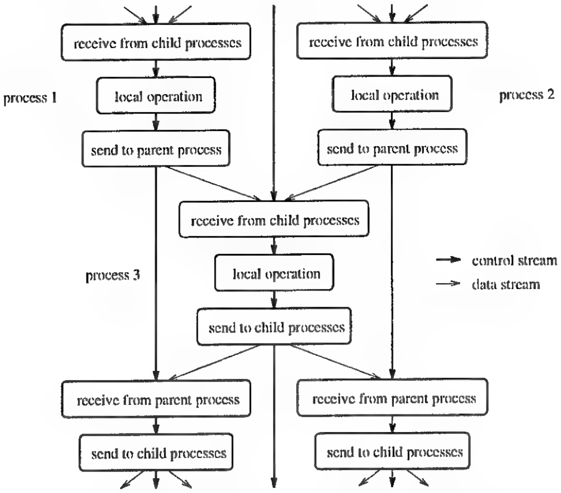


Fig. 6 Schematic flow chart for a global operation.

5. RESULTS

As a first result it should be noted that the parallel FLOWer code using the CLIC library meets all of the requirements stated above:

- The FLOWer code is fully portable, in sequential as well as in parallel mode.
- The effort spent for the development of the sequential FLOWer code and its predecessor CEVCATS was fully conserved.
- The effort needed for the parallelization was extremely low

The results obtained with this code are given in the following.

5.1 Performance Measurements

Since parallelization is a means of increasing the compute power for CFD applications, performance measurements were carried out on several platforms. With this not only the portability of the FLOWer code is demonstrated, but an assessment of different architectures is possible.

As test case the flow around a non-swept wing consisting of NACA 0012 airfoils was computed at M = 0.6 and $\alpha = 0^o$ (figure 7). Two different grids with 40000 and 320000 cells were used, respectively, that were subdivided into 1, 4 and 8 blocks in the small case and into 1, 4, 8, 16 and 32 blocks in the large case. Each block was of equal size and was mapped to one CPU on the parallel machines leading to an ideal load balance.

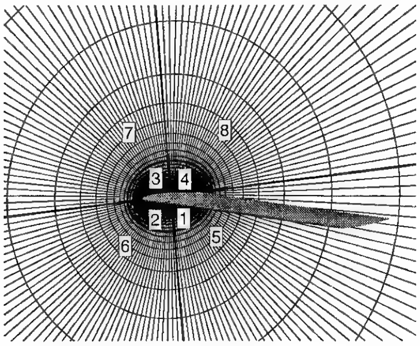


Fig. 7 NACA 0012 wing test case for performance measurements.

Figures 8 and 9 show the obtained computing times on various parallel and vector machines with respect to the time needed on a Cray C90 single processor. As can be seen, the single processor performance of the NEC SX-3 is hard to beat, even by parallel vector computers using up to 8 CPUs. On the other hand, the results show that parallel RISC processor architectures, as the IBM SP2 or the NEC Cenju-3, are able to compete with or even to outperform the Cray C90 single processor using a moderate number of 32 CPUs. The CM-5 and the Intel Paragon showed to have weaker single processors, so that they need many more CPUs in order to reach the performance of the other machines.

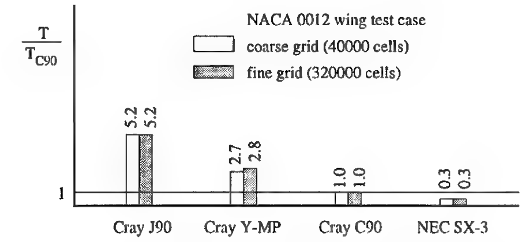


Fig. 8 Relative execution times on single processor vector computers

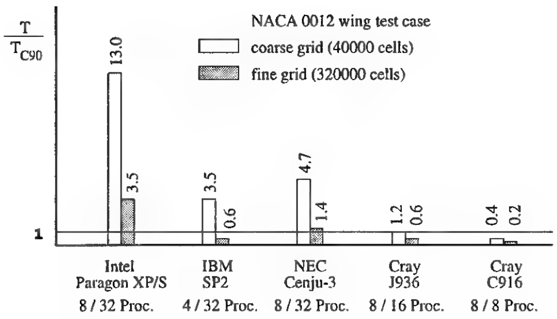


Fig. 9 Relative execution times on parallel computers

5.2 Speed-up for Aircraft Configurations

For evaluating the potential of the parallelization of the FLOWer code, speed-up measurements were carried out for a more realistic configuration. The inviscid flow around the generic DLR-F4 wing-body combination shown in figure 10 was computed on a grid consisting of approximately 410000 cells which was subdivided into 1, 4 and 8 equally sized blocks, respectively. For the conditions of Mach number M = 0.75 and incidence $\alpha = 0^o$, 35 W cycles involving 4 multigrid levels were performed on an IBM SP1 computer .

The results obtained for different communication systems available there are plotted as speed-up versus processor number in figure 11. As can be seen, PVM using an Ethernet connection restricts the processor number to be employed to only four indicating that workstation clusters based on the Ethernet are not suitable for parallel computations with the FLOWer code. The result can be markably improved, when replacing the Ethernet by the IBM high performance switch, but still the fastest runs were obtained using the IBM MPL/p communications system.

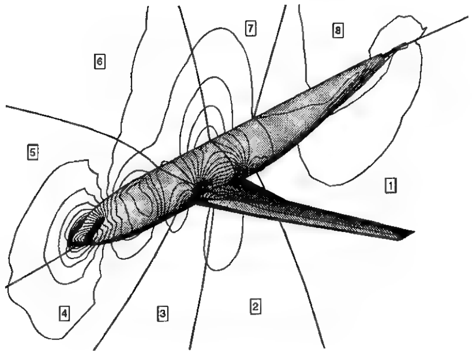


Fig. 10 Iso-Mach contours and block structure of the DLR-F4 wing-body combination (M = 0.75, $\alpha = 0^{o}$).

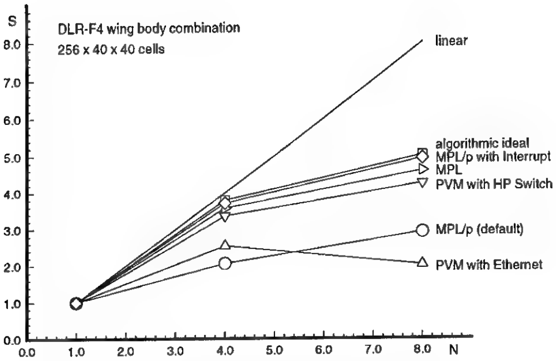


Fig. 11 Speed-up versus processor number for the DLR-F4 wing-body combination on IBM SP1.

What can be observed, is that even with the most powerful communciations systems on the IBM SP1, the acceleration obtained is considerably deviating from the linear speed-up. This effect is caused by an increase of operations due to the multiple computations of points at block interfaces.

Therefore, there is an upper limit for the maximum acceleration below the linear speed-up the which can be obtained from single processor computations of the multi block cases. This value is called algorithmic ideal speed-up an is defined as the ratio of computing times for the one block case and the multi block case multiplied with the number of processors that could be employed, i. e. the number of blocks:

$$S_{alg} = N_B \cdot \frac{t(N_B = 1)}{t(N_B)} \qquad (11)$$

The algorithmic ideal speed-up is also plotted in figure 11, and as can be seen, is reached to a degree of approximately 95% using the MPL/p system.

Of course, the decrease of the maximum obtainable speed-up reported above is not satisfactory, but on the other hand it is questionable, whether its value is meaningful for complex CFD problems at all. First of all, for speed-up measurements one would need problems that are small enough to be computed on a single CPU, in order to get a reference value. Secondly, the decrease in the maximum speed-up is only felt, because it was possible to compute a single block solution for the DLR-F4 wing-body combination. On the other hand the multi block cases were ideally load balanced, because all blocks were of equal size.

When dealing with more complex configurations, this will certainly not be the case. In such situations there will be several blocks from grid generation reasons which cannot be guaranteed to have all the same number of points. Therefore, more complicated test cases must be studied.

5.3 Parallel Computation of a Generic Aircraft

As a more realistic configuration, the DLR-ALVAST generic aircraft model carrying a high bypass engine [17] was computed at a Mach number of M = 0.75 and an incidence of $\alpha = 1.0^{o}$. The grid for this test case consists of about 575000 cells and is subdivided into 11 blocks the size of which is varying between 4096 and 87552 cells. This is a typical situation where neither load balancing nor single block computations are possible, both due to grid generation reasons. The configuration is shown in figure 12.

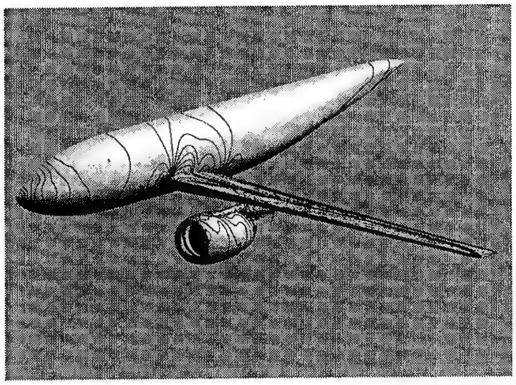


Fig. 12 ALVAST generic aircraft configuration. Iso-Mach lines at M = 0.75 and $\alpha = 0^{o}$.

In order to study the corresponding effects on the parallel performance, 50 W-cycles were performed mapping the 11 blocks to 1, 7 , 8 and 10 processors of an IBM SP2 respectively. The single processor result was obtained on a slightly more powerful wide node, whereas the parallel runs were obtained on weaker thin nodes.

As can be seen from figure 13, on 8 processors a speed-up of 6.6 can be gained, but a further increase does not lead to an improvement any more. This behavior is exactly what must be expected looking on the block structure and the mapping strategy of the CLIC library.

The work load per processor is determined by the number of grid points to be solved, and the largest number of points on any processor constitutes the total execution time of the parallel run. When mapping the 11 blocks to less than 11 processors, there will always be more than one block per CPU. Therefore, the CLIC library applies a mapping strategy that tries to distribute the blocks according to their size, so that the work load on the nodes is as equal as possible.

Up to 8 processors one is able to continously reduce the maximum grid size per CPU by simply mapping the largest block of the heaviest loaded node to an additional processor. But when employing 8 nodes, the maximum work load is determined by the absolutely largest block which of course cannot be reduced any further by mapping the block structure to more CPUs. Therefore, the minimum computing time or maximum speed-up, respectively, is to be obtained on 8 nodes and remains constant afterwards, as illustrated by figure 13.

Any further increase of the speed-up would require an additional blocking of the largest block which is planned to be automatically supported by the CLIC library in the future.

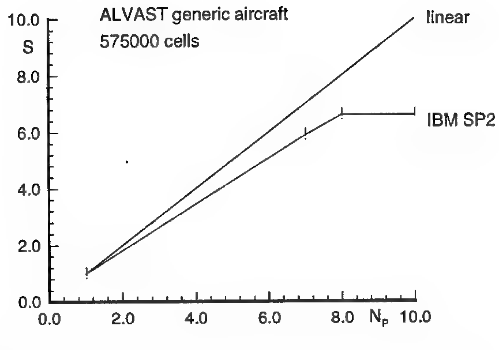


Fig. 13 Speed-up versus processor number for ALVAST generic aircraft configuration on IBM SP2.

5.4 Feasibility Study for Large Problems

Since parallelization is believed to be the appropriate method of tackling the future grand challenge problems in design aerodynamics, attempts must be made, in order to demonstrate the feasibility of this approach. Therefore, the viscous flow field around the DLR-F4 wing-body combination (compare figure 10) was computed on a grid generated by the Deutsche Airbus company consisting of 6.6 million grid points subdivided into 128 blocks of equal size. 800 multigrid cycles were performed on a 129 processor IBM SP2 (1 host + 128 nodes) which took less than three hours of response time (13 seconds per cycle). The convergence of the computation is given in figure 14 in terms of the logarithmic density residual versus the number of multigrid cycles.

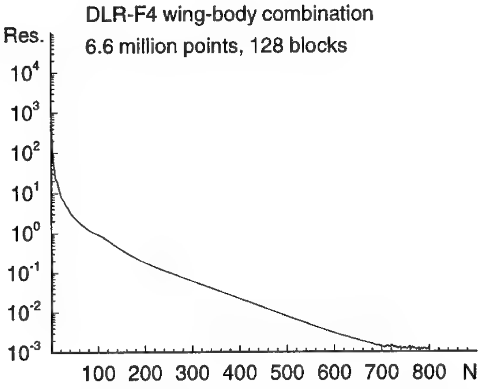


Fig. 14 Density residual versus number of multigrid cycles. DLR-F4 wing-body combination (6.6 million cells) on 129 processors of an IBM SP2.

A grid convergence study was carried out by repeating the computation on four grids each differing in the number of total points by a factor of 8. The result is given in figure 15 in terms of the total lift coefficient versus the scaled grid size. As one can see from an extrapolation of the development of the lift between the levels three and one, the large grid size of 6.6 million cells is necessary, in order to get the lift within an accuracy of one percent. Since the quality of the solution was spoiled by regions of highly distorted cells, a repetion of the study is planned with an improved grid.

Nevertheless, what is proven, is that such large scale problems can be treated with the parallel FLOWer code and that today‘s parallel hardware is already allowing such computations. Therefore, this study is a promising demonstration of the potential of parallel processing in CFD heading towards the solution of the aerodynamic grand challenges expected in the future.

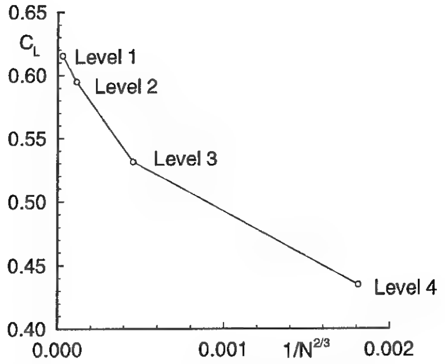


Fig. 15 Total lift coefficient versus scaled number of grid points. DLR-F4 wing body combination computed on 129 processors of an IBM SP2.

6. CONCLUSIONS

This paper shows, how the computational power of parallel architectures is exploited by the three-dimensional structured solver for complex flows FLOWer under the restricting demands for portability, conservation of the former development history and minimization of the parallelization effort.

The basic considerations to use the grid partitioning approach as parallelization strategy and to strictly separate communication and computation lead to the implementation of the message passing based portable CLIC-3D communications library supporting any high level operation occuring in typical partial differential equation solvers on structured meshes. With this library the parallelization meets all general requirements for the development of large codes in industrial use.

Performance measurements on a large variety of computers of different architecture demonstrate the comprehensive portability of the CLIC based FLOWer code and allow an assessment of today‘s hardware capability in CFD. Still the NEC SX-3 vector computer appeared to be the most powerful machine solving a standard benchmark problem, but with a moderate number of 32 RISC processors the IBM SP2 already outperforms a Cray C90 single processor, and a 32 processor NEC Cenju-3 is at least competitive.

Speed-up studies for a typical wing-body combination show that the communication system has a decisive influence on the achievable overall acceleration. It turns out, that Ethernet based workstation clusters communicating via PVM are not suitable to replace true parallel computers as far as performance is concerned.

Additionally, the maximum speed-up to be obtained is algorithmically limited by the grid partitioning strategy, because points at block interfaces are multiply computed increasing the total number of operations. But this drawback is only felt for simple problems, where a speed-up can still be measured and which are, therefore, far away from being a grand challenge.

Parallel computations of a generic aircraft consisting of a wing-body combination carrying a pylon with an engine demonstrate, that the complexity of today‘s problems in configuration aerodynamics can be tackled on a parallel computer. Speed-up measurements with respect to a multiblock single processor computation give satisfatory results, but also reveal the necessity for an automatic load balancing tool that allows to map an initial block structure to a higher number of processors than given blocks.

Finally, a Navier-Stokes computation of the flow field around a wing-body combination on a grid consisting of 6.6 million points on a 129 processor IBM SP2 outlines the potential of parallel processing in CFD for the future. It proves that high numbers of processors can be successfully handled in numerical aerodynamics and that parallelization, indeed, is a promising means for solving the grand challenge problems.

7. ACKNOWLEDGEMENTS

The work reported on here has been funded by the German Ministry of Research within the parallelization project POPINDA involving the following organizations:

Daimler Benz Aerospace Airbus Bremen, Daimler Benz Aerospace DASA-LM Manching/Ottobrunn, Deutsche Forschungsanstalt für Luft- und Raumfahrt Braunschweig, Gesellschaft für Mathematik und Datenverarbeitung St. Augustin, IBM Wissenschaftliches Zentrum Heidelberg.

The authors want to thank all contributors of these institutions.

8. LITERATURE

1. Holst, T. L., Salas, M., D., Claus, R. W., “The NASA Computational Aerosciences Program - Toward Teraflops Computing“, AIAA-92-0558, 1992

2. Agarwal, R. K., “Parallel Computers and Large Problems in Industry“ in “Computational Methods in Applied Sciences“, Elsevier Science Publishers, 1992

3. Kroll, N., Radespiel, R., Rossow, C.-C., “Accurate and Efficient Flow Solvers for 3D Applications on

Structured Meshes", AGARD FDP/VKI Special Course on Parallel Computing in CFD, 1995

4. Rossow, C.-C., "Berechnung von Strömungsfeldern durch Lösung der Euler-Gleichungen mit einer erwei-terten Finite-Volumen Diskretisierungsmethode", DLR-FB 89-38, 1989

5. Baldwin, B. S., Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows", AIAA-78-257, 1978

6. Jameson, A., Schmidt, W., Turkel, E., "Numerical simulation of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes", AIAA-81-1259, 1981

7. Kroll, N., Jain, R. K., "Solution of the Two-Dimensional Euler Equations - Experience with a Finite Volume Code", DFVLR-FB 87-41, 1987

8. Atkins, H., "A Multiple-Block Multigrid Method for the Solution of the Three-Dimensional Euler and Navier-Stokes Equations", DLR-FB 90-45, 1990

9. Eisfeld, B., Bleecke, H.-M., Kroll, N., Ritzdorf, H., "Parallelization of Block Structured Flow Solvers", AGARD-FDP/VKI Special Course on Parallel Computing in CFD, 1995

10. Rossow, C.-C., "Efficient Computation of Inviscid Flow Fields Around Complex Configurations Using a Multiblock Multigrid Method", Communications in Applied Numerical Methods 8, pp 737-747, 1992

11. Keyes, D. E, "Domain Decomposition: A Bridge Between Nature and Parallel Computers", ICASE Report No. 92-44, 1992

12. Lonsdale, G., Schüller, A., "Multigrid efficiency for complex flow simulations on distributed memory machines", Parallel Computing 19, pp 23-32, 1993

13. Hempel, R., Hoppe, H.-C., Supalov, A., "PARMACS 6.0 Library Interface Specification", GMD St. Augustin, 1992

14. Sunderam, V. S., "PVM, a framework for parallel distributed computing" in "Concurrency, Practice and Experience", Vol. 2(4), pp. 315 - 339, 1990

15. Message Passing Interface Forum, "MPI: A Message-Passing Interface Standard", University of Tennessee, 1994

16. Rossow, C.-C., Ronzheimer, A., "Investigation of Interference Phenomena of Modern Wing-Mounted High-Bypass-Ration Engines by the Solution of the Euler-Equations", AGARD-FDP Symposium on "Aerodynamic Engine/Airframe Integration", 7-10 october 1991

A Parallel Spectral Multi-Domain Solver Suitable for DNS and LES Numerical Simulation of Incompressible Flows

A. Pinelli and A. Vacca
School of Aeronautics, Polytechnic University of Madrid
Plaza Cardenal Cisneros 3
28040 Madrid, Spain

Abstract

In the present paper we introduce and discuss an efficient parallel algorithm for the spectral multi-domain solution of the incompressible Navier-Stokes equations. Firts, the algorithm is given in its basic form for the 2-dimensional case and, later on, a possible extension to 3-dimensional flows exhibiting a homogeneous (periodic) direction is proposed. The algorithm is validated both for its parallel performances, and its accuracy.

1 INTRODUCTION

In the last years domain decomposition methods have gained much attention in the CFD comunity. One of the most relevant features of such methods is concerned with the possibility of tuning the accuracy of the numerical discretization according to the expected behaviour of the solution in each subdomain. Consequently, subregions of flow field containing sharp boundary layer, can be enclosed within subdomains with high resolution, while low resolution can be assigned to subregions where smooth solutions can be expected.

These advantages can be fully exploited when discretizing the equations with spectral methods which guarantee a fast decay of the error with the number of the nodes, termed as "spectral accuracy".

On the other hand domain decomposition methods might provide a natural stabilization strategy for the spectral discretization which is a "central one" in nature. In fact the local cell Peclet number can be locally diminished by reducing the mesh spacing within the critical subdomain, without the introduction of any particular stabilization procedure.

From the computational point of view, the domain decomposition techniques is well suited for parallel computing, even if in practical case several difficulties arises whenever good performances have to be reached [1].

In the first part of the present paper, a parallel algorithm for the solution of the bidimensional incompressible Navier-Stokes equations is presented. After a brief introduction of the time splitting scheme used for the time discretization of the unsteady incompressible Navier-Stokes equations, the attention will be mainly focused on the the spectral multidomain approach and on its parallel features. Performance results concerning the parallel implementation on two different MIMD parallel architectures will be presented. The second part of the paper is concerned with the application of the algorithm to three dimensional unsteady problems.

2 NAVIER-STOKES EQUATIONS AND TIME SPLITTING SCHEME

When the incompressible Navier-Stokes equations

$$\frac{\partial U}{\partial t} + \frac{1}{2}\left(U \cdot \nabla U + \nabla \cdot (UU)\right) = -\nabla p + \frac{1}{Re}\,\Delta\, U \quad (1)$$

$$\nabla \cdot U = 0 \quad (2)$$

are solved by means of a projection method [2], with the diffusive terms treated in an implicit fashion [3], the time stepping procedure consists in a cascade of scalar elliptic kernels, to be solved at each time step. Namely two (for the two-dimensional equations) Helmohltz problems for the inversion of the diffusive part, and a Poisson problem for the pressure need to be solved at each time step. It is then clear that, in order to achieve a globally efficient algorithm, it is of fundamental importance to tackle effectively the mentioned scalar problems. For the sake of completeness in the following the adopted fractional step scheme (i.e. Van Kan's pressure correction method [4])is given

$$\frac{\hat{U} - U^n}{\Delta t} - \frac{1}{2Re}\,\Delta\left(\hat{U} + U^n\right) = -\nabla p^n - \frac{3}{2}\mathcal{L}\left(U^n\right) + \frac{1}{2}\mathcal{L}\left(U^{n-1}\right) \quad (3)$$

$$\hat{U}|_{\partial\Omega} = U\left((n+1)\,\Delta t\right) \quad (4)$$

$$\frac{U^{n+1} - \hat{U}}{\Delta t} + \frac{1}{2}\nabla\left(p^{n+1} - p^n\right) = 0 \quad (5)$$

$$\nabla \cdot U^{n+1} = 0 \quad (6)$$

where $\mathcal{L}(U)$ represents the advective term $\frac{1}{2}\left(U \cdot \nabla U + \nabla \cdot (UU)\right)$.

In the first step, a non physical intermediate velocity field $\hat{U}$ is computed. In fact, $\hat{U}$ does not satisfy the incompressility condition. Then in the second step $\hat{U}$ is projected onto the divergence free space to get an adeguate velocity appressimation of U^{n+1}.

The scheme with the given boundary conditions is nothing else then a second order Crank-Nicolson Adams-Bashforth scheme with an $\mathcal{O}\left(\Delta t^2\right)$ deviation in the tangent direction of the boundary. By applying the divergence operator to (6), it turns out that the latter is equivalent to

$$\Delta\left(p^{n+1} + p^n\right) = \frac{2}{\Delta t}\nabla \cdot \hat{U} \quad (7)$$

$$\frac{\partial p^{n+1}}{\partial n}|_{\partial\Omega} = 0 \quad (8)$$

$$U^{n+1} = \hat{U} - \frac{\Delta t}{2}\nabla\left(p^{n+1} - p^n\right) \quad (9)$$

In the next section the attention will be focused on the way each scalar elliptic problem has been tackled in the framework of a spectral multidomain discretization.

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

3 SPACE DISCRETIZATION

In the present work, a Legendre spectral collocation technique coupled with a domain decomposition method has been used for the space discretization of the differential equations. Additional references can be found in ([6], [5]) for the projection decomposition method, and in ([7]) for the spectral approximation method.

3.1 Elliptic terms

The following problem, rappresentative of one of the elliptic scalar problems mentioned in the previous section, is considered hereafter:

$$\begin{aligned} -\Delta u + \alpha u &= f \quad \text{in } \Omega, \quad f \in L^2(\Omega) && (10)\\ u &= 0 \quad \text{on } \partial\Omega && (11)\end{aligned}$$

where α is a real constant ≥ 0, and where Ω is an open connected set $\Omega \subset \Re^2$; in particular, $\overline{\Omega} = \cup_{i=1}^{N} \overline{\Omega_i}$ with $\overline{\Omega_i}$ is a closed rectangle having either common side or common vertex with each neighbour; $\alpha \geq 0$ is either identically equal to zero (i.e., for the Poisson problem related with the pressure) or is equal to $2/\Delta t Re$ (i.e., for one of the momentum equations), and the equivalent weak formulation of (10), (11) is:

$$\text{find } u \in H_0^1(\Omega) \quad \text{such that}$$
$$l(u,v) = (f,v)_{L^2(\Omega)} \; \forall\, v \in H_0^1(\Omega) \quad (12)$$

where $H_0^1(\Omega)$ is the real Hilbert space defined as follows:

$$H_0^1(\Omega) \equiv \{u \in L^2(\Omega) : \frac{\partial u}{\partial x_1} \in L^2(\Omega) \quad (13)$$

$$\text{and } \frac{\partial u}{\partial x_2} \in L^2(\Omega), \; u|_{\partial\Omega} = 0\} \quad (14)$$

equipped with the scalar product:

$$l(u,v) = \int_\Omega (\nabla u \cdot \nabla v + \alpha u v) d\Omega \qquad \forall u, v \in H_0^1(\Omega) \quad (15)$$

Following the classical domain decomposition technique problem (12) is decoupled into a set of problems within each subdomain plus an additional problem at the interfaces Γ:

$$\Gamma = (\Omega \backslash \Omega_0) \backslash \partial\Omega \quad \text{with} \quad \Omega_0 = \cup_{i=1}^{N} \Omega_i \quad (16)$$

Let $H_0^{1/2}(\Gamma)$ be the completion of the normed vector space S defined as:

$$\begin{aligned} S &\equiv \{z \in C^0(\Gamma) : \exists \phi \in C_0^\infty(\Omega) \text{ such that } z = \phi_\Gamma\}\\ \|z\| &= \inf_{\substack{\phi \in C_0^\infty(\Omega)\\ \phi_\Gamma = z}} \|\phi\|_{H_0^1(\Omega)} && (17)\end{aligned}$$

where ϕ_Γ is the restriction of ϕ on Γ.
The linearity and continuity of the operator

$$\phi \in C_0^\infty(\Omega) \rightarrow \phi_\Gamma \quad (18)$$

into $H_0^{1/2}(\Gamma)$ and the fact that $C_0^\infty(\Omega)$ is dense in $H_0^1(\Omega)$ leads to the existence and uniqueness of a linear and continuous operator γ (trace operator) from $H_0^1(\Omega)$ onto $H_0^{1/2}(\Gamma)$ defined as

$$\gamma\phi = \phi_\Gamma \quad \forall \phi \in H_0^1(\Omega) \quad (19)$$

The γ operator allows to identify two closed mutually orthogonal subspaces

$$K \equiv \ker(\gamma) = \{u_0 \in H_0^1(\Omega) : \gamma u_0 = 0\} \quad (20)$$

where $\ker(\gamma)$ is the kernel of operator γ, and its orthogonal complement $K^\perp$ is defined as:

$$K^\perp \equiv \{\tilde{u} \in H_0^1(\Omega) : l(\tilde{u}, v_0) = 0 \quad \forall\, v_0 \in K\} \quad (21)$$

Therefore, the solution $u \in H_0^1(\Omega)$ of problem (12) can be uniquely decomposed as

$$u = u_0 + \tilde{u}, \quad u_0 \in K \quad \text{and} \quad \tilde{u} \in K^\perp \quad (22)$$

Since the restriction γ_0 of the operator γ to $K^\perp$ is an isometric isomorphism between $K^\perp$ and $H_0^{1/2}(\Gamma)$ it follows that

$$\forall\; \tilde{u} \in K^\perp \quad \exists!\, \psi \in H_0^{1/2}(\Gamma) : \quad \tilde{u} = \gamma_0^{-1}\psi \quad (23)$$

Identity (22) can be reformulated as:

$$u = u_0 + \gamma_0^{-1}\psi \quad \text{with} \quad u_0 \in K \quad \text{and} \quad \psi \in H_0^{1/2}(\Gamma) \quad (24)$$

Thus, problem (12) can be easily proven to be equivalent to the set of the two following ones:
Problem (P1): find $u_0 \in K$ such that:

$$l(u_0, v_0) = (f, v_0)_{L^2(\Omega)} \;\; \forall\, v_0 \in K \quad (25)$$

Problem (P2): find $\psi \in H_0^{1/2}(\Gamma)$ such that:

$$l(\gamma_0^{-1}\psi, \gamma_0^{-1}z) = (f, \gamma_0^{-1}z)_{L^2(\Omega)} \forall z \in H_0^{1/2}(\Gamma) \quad (26)$$

Problem P1 is nothing else than the solution of N decoupled elliptic problems with homogeneous Dirichlet boundary conditions on both $\partial\Omega$ and Γ. To build its discrete conterpart, a standard Legendre collocation method has been used ([7]). To this end, the unknowns are decomposed into a series of Legendre polynomials:

$$u(x,y) = \sum_{k=1}^{N_x} \sum_{l=1}^{N_y} u_{k,l} L^k(x) L^l(y), \quad (27)$$

where L^k is the k^{th} Legendre polynomial. Likewise, the function v is decomposed into a series of Lagrange polynomials constructed on the Gauss-Lobatto nodes.

$$v(x,y) = \sum_{k=1}^{N_x} \sum_{l=1}^{N_y} v_{k,l} La^k(x) La^l(y), \quad (28)$$

where La^k is the k^{th} Lagrange polynomial for which $La^k(x_i) = \delta_{k,i}$. By taking into account the expression of u and v and by replacing the scalar product $l(.,.)$ by its discrete counterpart, the differential problem reads:

$$\left\{ \begin{array}{l} \text{find } u_{k,l}, 1 \leq k \leq N_x \,,\; 1 \leq l \leq N_y \text{ such that} \\ \sum_{k,l} [-(\frac{\partial^2 u}{\partial x_m^2})_{k,l} + \alpha u_{k,l} - f_{k,l}] La^i(x_k) La^j(y_l) \omega_k \omega_l = 0 \;\; \forall i, j \end{array} \right. \quad (29)$$

where ω_k are the Gauss Lobatto weights for the quadrature. Using the definition of Lagrange polynomials ($La_j(x_i) = \delta_{ij}$), the disretized equations become:

$$\left\{ \begin{array}{ll} \text{find } u_{k,l}, & 1 \leq k \leq N_x \,,\; 1 \leq l \leq N_y \\ \text{such that} & \frac{\partial^2 u_{k,l}}{\partial x_j^2} + \alpha u_{k,l} = f_{k,l} \end{array} \right. \quad (30)$$

An efficient procedure to solve the given algebraic problem will be given in the next section.
As concern problem P2, if $\{\xi_i\}$ $i = 1, .., \infty$ is a set of linearly independent functions which constitute a base for $H_0^{1/2}(\Gamma)$, then the discrete version of problem P2 reads as:

$$l(\gamma_0^{-1} \sum_{i=1}^{M} a_i \xi_i, \gamma_0^{-1}\xi_j) = (f, \gamma_0^{-1}\xi_j)_{L^2(\Omega)} \quad \forall\, j = 1, .., M \quad (31)$$

Typically M corresponds exactly to the number of points on the interface. To set up an algebraic equivalent of (31) the operator γ_0^{-1} should be explicitly formulated. In practice, the operator γ_0^{-1} is never required if an iterative procedure is introduced to solve problem P2. To illustrate this point, it must be remarked that $\tilde{u}^k \in K^\perp$ must satisfy the orthogonality condition:

$$l(\tilde{u}^k, v_0) = 0 \quad \forall\, v_0 \in K \tag{32}$$

which corresponds to the solution of N elliptic problems (25) with Dirichlet boundary conditions: homogeneous on $\partial\Omega$ and to be iteratively determined on Γ.
To provide at each iteration k the condition on Γ for problem (32) the Green's formula is applied to (31)

$$R^k = \int_\Gamma \frac{\partial \tilde{u}^k}{\partial n}\xi_j d\Gamma - \int_\Gamma \frac{\partial u_0}{\partial n}\xi_j d\Gamma \quad \forall\, j = 1,..M \tag{33}$$

where $\tilde{u}^k = \gamma_0^{-1}\sum_{i=1}^M a_i^k \xi_i$ is the solution at iteration k of problem (32), where $\frac{\partial}{\partial n}$ represents the jump of the normal derivatives on Γ. R^k is the residual at iteration k, from which the updating of the boundary value $\tilde{u}^{k+1}|_\Gamma$ can be obtained within the chosen iterative procedure.
The convergence rate of the iterative procedure strongly depends on the choice of the basis $\{\xi_i\}$ [8]. For the present work the basis functions proposed by Ovtchinnikov [8] have been used. These constitute a nearly optimal basis, in the sense that the condition number of system (31) is bounded by a constant independent of M, where M is the dimension of the subspace of $H_0^{1/2}(\Gamma)$ generated by $span\{\xi_i\}$ $i = 1,..,M$.
In view of the character of the algebraic problem (symmetric positive defined) the conjugate gradient has been employed to solve problem (31).

3.2 Solution procedures for multiple problems

When multiple solutions for an elliptic problem of the form (10, 11) are required (i.e., within a fractional step time advancement), it turns out to be much more efficient to invert, once for all (in a pre-processing stage), the abstract operator $\mathcal{S}$ handling the interface unknowns.
To introduce the method let us reconsider problem (10, 11). With reference to the previous section, we reconsider the same differential problems: problem P1 (25) and the the differential problem leading to the solution on the interface (P2), here given in the following abstract form:

$$\mathcal{S} a_k \;=\; b_k \tag{34}$$

Where the a_k's refers to the Galerkin coefficients of the solution on the interface, and the b_k's are the Galerkin coefficients of the jump of the normal derivatives produced by the solution of the N problems P1.
Let us now consider the M problems (see 31):

$$\mathcal{S} a_k \;=\; \delta_{ij} \tag{35}$$

Meaning problems with a jump of the normal derivatives leading to a unitary Galerkin coefficient i and zero values for all the other coefficients j, $(j \neq i)$. Succesive inversions, through the iterative procedure outlined in the previous section, allow for constructing by columns the operator $\mathcal{S}^{-1}$. The latter might, then, be considered as a capacitance Galerkin matrix that applied to the Galerkin coefficients of the computed normal derivatives jumps (problem P1) release the coefficients of the solution on the interface to guarantee a zero weak normal derivative jumps between subdomains. It is also remarked that matrix $\mathcal{S}^{-1}$ is simmetryc because obtained from the discretization of a self-adjoint problem (33). Of course this is a nice property leading to an evident storage reduction.

A final remark concerns the importance of achieving an efficient technique to invert the decoupled Dirichlet problems (P1). To this end, we make use of a modified matrix diagonalization approach [9]. The Legendre collocation approximation to one of the mentioned sub-problems migh be re-written as:

$$UD + D^T U + \alpha I U \;=\; F \tag{36}$$

where D is the collocated Lagrange second derivative matrix acting on the subdomain internal nodes, U is the unknow matrix ordered by rows, and F is a modified right hand side matrix keeping into account the effects of the boundary values. First, we determine the eigenvalues of D, its left and right eigenvector system (ordered by columns) and the respective inverses.

$$E_r^{-1}\; D\; E_r \;=\; \Lambda \tag{37}$$

$$E_l^{-1}\; D^T\; E_l \;=\; \Lambda \tag{38}$$

Matrices E_r, E_l, E_r^{-1}, E_l^{-1} and the diagonal eigenvalue matrix Λ are computed and stored in a pre-processing stage. Indicating with $\hat{U} = E_r^{-1}\; U\; E_l$ and with $\hat{F} = E_r^{-1}\; F\; E_l$ we invert the diagonalised problem:

$$\Lambda\hat{U} \;+\; \hat{U}\Lambda \;-\; \alpha\hat{U} \;=\; \hat{F} \tag{39}$$

and recover the final solution as:

$$U \;=\; E_r \hat{U} E_l^{-1} \tag{40}$$

Having solved the eigenvalue problem in a pre-processing stage, the recursive solution cost turns out to be order n^3 operations, n being the number of nodes used to discretized each direction within a single subdomain.

3.3 Projection step treatment

If each single differential problem is tackled with the algorithm described in the previous section, at the end of each time step the solution is equivalent to one, hypotetically achieved by solving the whole domain at once.
The last statement requires some comments. When finite dimensional approximation of the space where the solution is sought are considered numerical problem might arise within the fractional step algorithm in the interface neighbouring regions. In particular when the projection step (5) is considered, a straigh use of the results obtained with the present multidomain method leads a discontinuous value of the divergence free velocity field along the interfaces.
From the numerical point of view, these discontinuities, even if limited to a set of measure zero (Γ) might introduce an artificial "numerical boundary layer" that the whole time integration procedure cannot damp out and that might lead to catastrophic instabilities. To avoid such a drawback two solutions are possible. The first one relies upon increasing the dimension of the approximation subspaces to reduce the jumps at the interface. The second one consists in replacing the gradient of the pressure in equation (5) with an equivalent function in $L^2(\Omega)$, which differs from the original one only along sets of measure zero. In particular, the gradient $\mathcal{Q} = \nabla(\phi^{n+1} - \phi^n)$ of the solution, achieved by solving (7) with the previously outlined multi-domain spectral method is substituted with the vector function $\mathcal{G}$ defined as:

$$\mathcal{G}_i(x,y) = \begin{cases} \mathcal{Q}_i(x,y) & \forall\,(x,y) \in \overline{\Omega}\backslash\Gamma \\ \frac{1}{\omega_j^r + \omega_j^s}[\mathcal{Q}_i^r \omega_j^r + \mathcal{Q}_i^s \omega_j^s] & \forall\,(x,y) \in \Gamma_{rs} \end{cases} \tag{41}$$

$\forall\, i,j = 1,2$ component.
where $\Gamma_{rs} = \overline{\Omega}_r \cap \overline{\Omega}_l$, ω_j^r (ω_j^s) is the Gauss-Legendre quadrature weight along Γ_{rs} (either j or i) corresponding to the node (x,y) in the subdomain $\overline{\Omega}_r$ $(\overline{\Omega}_s)$ and $\mathcal{Q}^r$ $(\mathcal{Q}^s)$ is the restriction of $\mathcal{Q}$ in $\overline{\Omega}_r$ $(\overline{\Omega}_s)$ evaluated in (x,y).

3.4 Accuracy tests

To test the accuracy of the proposed spectral multi-domain algorithm we have considered the classical Taylor-Green analytical test case for the 2-dimensional incompressible Navier-Stokes equations:

$$u(x,y) = -\cos(\pi x)\ \sin(\pi y) e^{-t/2\pi^2} \qquad (42)$$

$$v(x,y) = \sin(\pi x)\ \cos(\pi y) e^{-t/2\pi^2} \qquad (43)$$

$$p(x,y) = -1/4\ (\cos(2\pi x) + \cos(2\pi y)) e^{-t\pi^2} \qquad (44)$$

on the domain $\Omega = [0,2] \times [0,2]$. The following set of boundary conditions have been applied:

- on the edges $x = 0$ and $x = 2$ homogeneous Dirichlet conditions for u and homogeneous Neumann conditions for v.
- on the edges $y = 0$ and $y = 2$ homogeneous Dirichlet conditions for v and homogeneous Neumann conditions for u.

The tests have concerned both time and space accuracy. The latter has been measured imposing an extremely small value for the time step. Different configurations have been considered and the error has always been measured according with the $L^2(\Omega)$ norm. The following table, showing the results of different tests with different domain partitioning configurations, summarizes the accuracy measurements both for one of the velocity components and for the pressure.

Num.	Nodes	Total	Error	Error
doms	per dom.	nodes	L2 u	L2 p
1	8	64	$.6 \times 10^{-9}$	$.2 \times 10^{-4}$
1	11	121	$.2 \times 10^{-13}$	$.2 \times 10^{-7}$
1	14	196	$.7 \times 10^{-17}$	$.8 \times 10^{-9}$
4	8	256	$.9 \times 10^{-14}$	$.6 \times 10^{-9}$
4	11	484	$.8 \times 10^{-15}$	$.6 \times 10^{-12}$
4	14	784	$.6 \times 10^{-17}$	$.8 \times 10^{-17}$

From the given results, the accuracy of the solution is quite evident. It is remarked that the convergence for the pressure is lower than for the velocity, but nevertheless, still spectral. In order to measure the time accuracy of the present scheme, we considered the same test case with a prescribed discretization in space (4 subdomains 14 × 14 nodes each) sufficient to deliver optimal spatial accuracy. In the following table we present the relative $L^2(\Omega)$ norm of the velocity error achieved after 1 time unit.

time step size	Relative L2 x-component velocity error
0.1	$.4 \times 10^{-1}$
0.01	$.5 \times 10^{-3}$
0.001	$.3 \times 10^{-5}$
0.0001	$.5 \times 10^{-7}$

From the results it turns out that the adopted scheme is second order in time, at least for the velocity.

4 PARALLEL IMPLEMENTATION

As concerns the parallel implementation of the given algorithm, we have used a slightly modified version of master-slave computational model. In particular, the major difference with respect to the classical model is that our master actively cooperates with the slaves during the calculation phase, while in the standard version, the master is only demanded to distribute initial data and to gather the results. In our implementation, the activities are shared between master and slaves as follow. At the beginning of the computation, the master process calculates the guess values for the Dirichlet problems. These values are then trasmitted to the slave processes: each of the slaves solves the Dirichlet problems for the assigned domains; it should be noted that, in this case, the domain decomposition (which allow the slaves to operate in parallel) derives directly from the multi-domain approach. After this first phase, the slaves transmit the calculated values at the domain interfaces to the master, which calculates the new values by applying a Conjugate Gradient algorithm, and communicates these values to the slaves for the next iteration. The main causes of inefficiency in using parallel architectures are an uneven load-balancing and the communication overheads. In general, the multidomain technique can generate load balancing problems because the size and/or computation of blocks can widely differ; however, in our case each domain has the same number of points. Thus, if the number of domains is a multiple of the number of processor, we obtain an optimal load balancing. The communication overheads is mainly related to the Conjugate Gradient algorithm: at each time iteration, data need to be exchanged between processors containing adjacent domain interfaces and the master processor. Because of the sequential flow of these activities, it is not possible to overlap computation and communication, so the time spent for these communications can represent a not negligible part of the overall computing time.

The parallel version of the code has been developed for message passing environments. In particular, the code has been written in Fortran 77 plus PVM 3.3 communication primitives. In order to meet the goal of overlapping computation and communication, non-blocking communication primitives have been used. Note that the parallelism is exploited only among slaves: the master and the slaves cannot operate in parallel. Anyway, as the great part of the computation is demanded to the slaves, the obtained performances on various homogeneous parallel systems are quite good.

4.1 Performance evaluations

For the tests, we have used two different parallel machines. The first is a CONVEX C210-MPP0 with a vector processor and 4 scalar processor HP 730 connected via FDDI. The second machine is a MEIKO CS2 with 18 super-Sparc processors connected through a switching network. Both the machine are distributed memory MIMD parallel computers. The tests have been performed by using a number of domain multiple of the number of processors used, so that load balancing is guaranteed. Hence, the cause for the loss of efficiency are the time t_c spent for the communication and the idle time t_w of the slaves waiting for the master results. Note that, while the time t_w is indepenedent of the number N of processors, the time t_c increases according to N: so, for a given problem, a linear decrease of the efficiency is expected.

In figg. (1-4) the results obtained on the Meiko machine are shown. Note that the values of efficiency are quite good, expecially when the number of points for each domain increases. Moreover, when the number of processor grows the efficiency linearly decrease, as expected.

Figures (5,6) shows a comparison of the results obtained for both the Meiko CS2 and the Convex MPP0 machines. It should be noted that the Convex machine performs better than Meiko when two processors are used; on contrary, by increasinging the number of processors the performances of the Meiko are better. This behaviour is essentially related to the different characteristics of the interconnection networks; the FDDI network of the Convex allows very fast communication between two processors at time (the optical fiber is a common shared resource). On the other hand, the CS2 switching network allows to simultaneously execute different communications, so reducing the overall communication time (as matter of fact, also the presence of properly designed communication

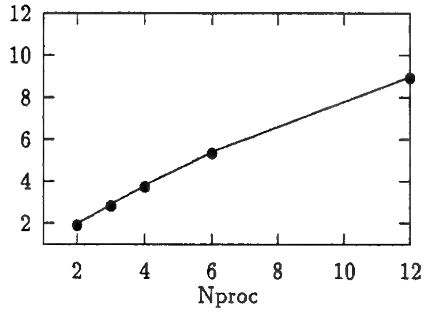


Fig. 1: 12 domains with 15 × 15 nodes; speed up

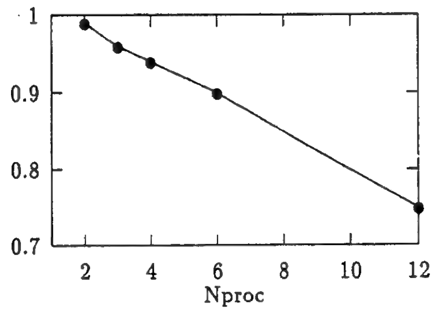


Fig. 2: 12 domains with 15 × 15 nodes; efficiency

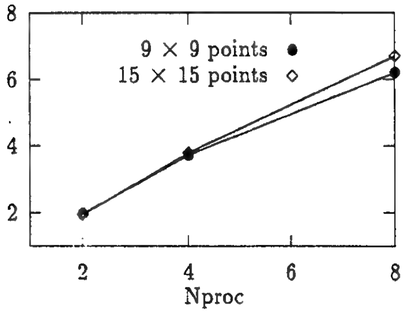


Fig. 3: 16 domains; speed up

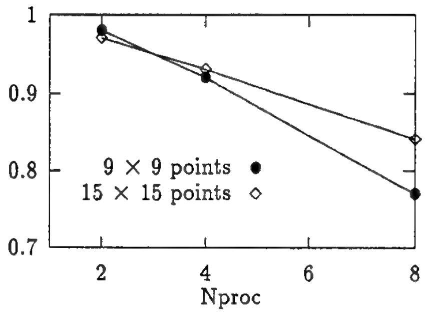


Fig. 4: 16 domains; efficiency

processors which handle the communication on behalf of the sparc processor has to be taken into account).

To further reduce the computing time, we have also used heterogeneous systems. In fact, whenever the execution of different tasks constituting the same program is strictly sequential, heterogeneous processing can help in enhancing performance by placing a task on the most suitable machine for that task. To this goal, tests have been performed by placing the master process on a vector computer for a more efficient calculation, and the slave processes on a parallel homogeneous system with scalar processors.

However, in our case the time spent by the master is a negligible part of the total comuputing time; so, the test performed by using an heterogeneous environment have shown no appreciable improvements.

5 3-DIMENSIONAL EXTENSION

In this section, we present a method extension which allows for the simulation of three-dimensional flows with one periodical direction. For this class of flows it is possible to take advantage of the classical Fourier decomposition of the flow variables in the periodical direction. This choice leads to reduce all the three-dimensional scalar differential problems in the physical space (momentum equations and pressure correction equation) into a sequence of two-dimensional scalar differential problems in terms of the transformed variables. Once the two-dimensional problems are set up, it is possible to take advantage of the given multi-domain solution method to solve them efficiently.

In particular, let

$$u_i^n(x,y,z) = \sum_{k=-N/2}^{N/2-1} \tilde{u}_{i,k}^n(x,y)e^{Ikz}, \quad i = 1,2,3, \qquad (45)$$

$$p^{n+1}(x,y,z) = \sum_{k=-N/2}^{N/2-1} \tilde{p}_k^{n+1}(x,y)e^{Ikz}, \qquad (46)$$

$$\hat{u}_i(x,y,z) = \sum_{k=-N/2}^{N/2-1} \tilde{\hat{u}}_{i,k}(x,y)e^{Ikz}, \quad i = 1,2,3, \qquad (47)$$

and

$$\delta\hat{u}_i(x,y,z) = \sum_{k=-N/2}^{N/2-1} \delta\tilde{\hat{u}}_{i,k}(x,y)e^{Ikz}, \quad i = 1,2,3, \qquad (48)$$

with $I = \sqrt{-1}$. Applying the same methodology as for the 2-dimensional case, the 3-D dimensional algorithm can be reformulated as:

For every $n = 0, 1, ..$ (n being the time counter)

1 For $i = 1,2,3$, solve for $\tilde{\hat{u}}_{i,k}$ (the predicted velocity field) the momentum equations, for $k = -N/2, ..., N/2-1$:

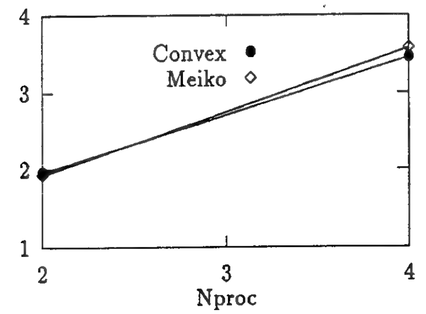


Fig. 5: 3 domains with 11 × 11 nodes; speed up

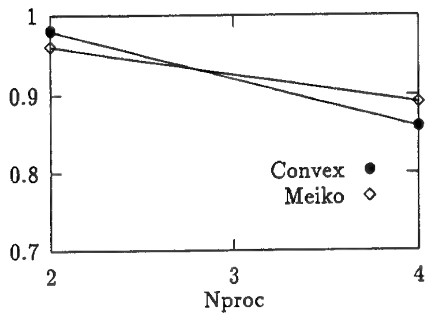


Fig. 6: 3 domains with 11 × 11 nodes; efficiency

$$(-\theta\frac{\partial^2}{\partial x_l^2}+\theta k^2+\frac{Re}{\Delta t})\delta\tilde{\hat{u}}_{i,k}=r\tilde{h}s_{i,k},\quad l=1,2, \qquad (49)$$

2 Solve for $\tilde{p}_k^{n+1}$ the pressure correction, for $k = -N/2,...,N/2-1$:

$$(-\frac{\partial^2}{\partial x_l^2}+k^2)\tilde{p}_k^{n+1}=-\frac{2}{\Delta t}\frac{\partial\tilde{\hat{u}}_{l,k}}{\partial x_l}+\frac{Ik}{\Delta t}\tilde{\hat{u}}_{3,k}+(-\frac{\partial^2}{\partial x_l^2}+k^2)\tilde{p}_k^n \qquad (50)$$

3 For $i = 1,2,3$, update the velocity field, for $k = -N/2,...,N/2-1$:

$$u_{i,k}^{n+1}=\tilde{\hat{u}}_{i,k}-\Delta t\begin{cases}\frac{\partial(\tilde{p}_k^{n+1}-\tilde{p}_k^n)}{\partial x_i} & if\ i=1,2\\ Ik.\tilde{p}_k^{n+1} & otherwise\end{cases} \qquad (51)$$

The subscript l has been introduced to stress the fact that the collocated derivatives are computed in the two non-periodical directions only. The term $r\tilde{h}s_{i,k}$ represents the k^{th} mode of the transform of the right-hand-side of the i^{th} momentum equation. The treatment of the boundary conditions is straightforward and does not introduce any supplementary difficulty. Despite its apparent complexity, this algorithm presents the advantage that all the computations of the elliptic terms take place in the transformed space (for the periodic direction) leading to the full exploitations of the 2-dimensional algorithm.

5.1 A 3-D test case

To validate the proposed 3-D algorithm we have considered a direct numerical simulation (DNS) of a low Reynolds number fully turbulent channel flow. This flow ([10]) might be considered periodic both in stream and spanwise direction if the dimensions of the computational box are made large enough. In the present case we took as Fourier direction the streamwise one, while, to impose periodicity spanwise we imposed the edges of the subdomains to be neighbours one with the other. All the lenghts have been made non-dimensional with the channel half-height, and the velocity has been non-dimensionalized with the center-line velocity. With this selection the Reynolds number $Re = U_c\ h/\nu$ has been fixed to the value of 6000 and the computational box had dimensions 2, 2, .8 in streamwise, normal to the wall and spanwise directions respectively . The grid configuration in a section normal to the mean flow is displayed in figure (7).

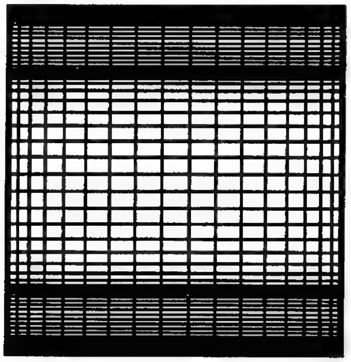


Fig. 7: Grid configuration in the normal plane

Five subdomains, the first and the latter selected to embedd the wall sublayer, are used. Each subdomain contains 20 × 20 nodes, while in the Forier direction 24 modes are employed. The present case has been run on a IBM RS6000 360H workstation with about $100Mflops$ peak performance. The cpu required for each full time iteration is of about 4.5 seconds when the Galerkin capacitance matrix is computed and stored in a pre-processing stage.

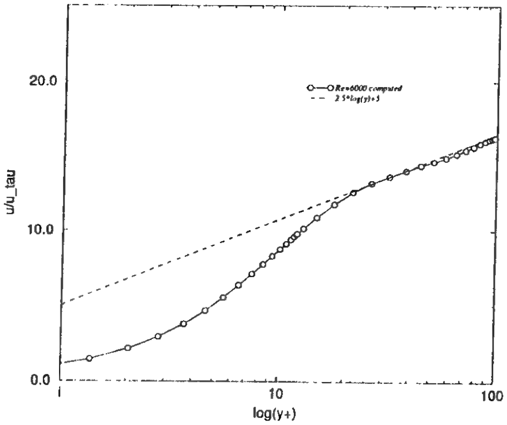


Fig. 8: Mean stream-wise velocity near tne wau

After having reached a statistical steady state we measured some typical turbulent value to assess the quality of the obtained results. In figure (8) we compare the obtained velocity profile with the logarithmic wall law ($u/u_{tau} = 2.5\ log(y^+) + 5$). In figure (9) the computed turbulence intensities are compared both with the experimental ones of Wei and Willmarth ([11]) at Reynolds 3850 and with the ones predicted by the DNS of Jiménez and Moin ([10]) at Reynolds 5000.

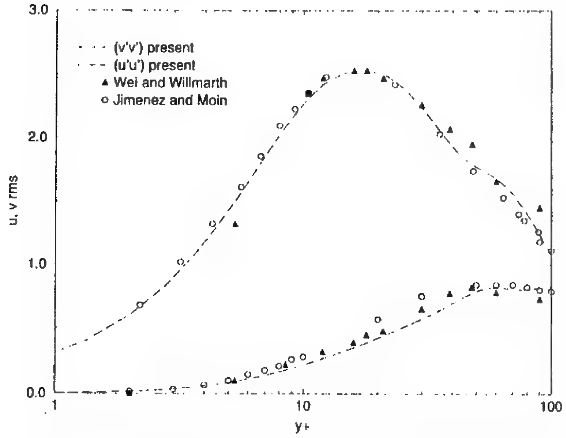


Fig. 9: Near the wall turbulence intensities in wall coordinates

Finally, an instantaneous velocity field in a plane normal to the mean stream is displayed in figure (10). The agreement of the data is acceptable and paves the way for simulation of more complex turbulent flow configurations. Indeed when more points are necessary the actual code might allow for a full exploitation of MIMD computers architectures.

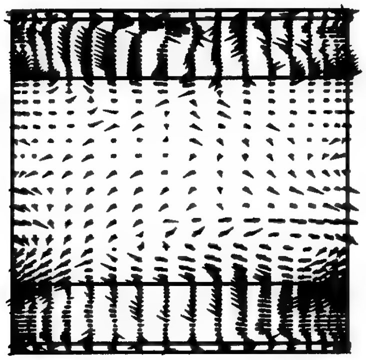


Fig. 10: Instantaneous normal plane velocity field

6 CONCLUSION

The present work has been concerned with the solution of the unsteady incompressible Navier-Stokes equations, using a high order collocated spectral multi-domain method. The rationale behind the choice and development of the method is given both by the possibility of coupling the potential high accuracy of spectral methods with the flexible framework offered by multi-domains methods, and with the natural way in which a parallel implementation of the present algorithm can be achieved.

In particular, we have shown how the developed algorithm allows for the solution of completely independent and balanced sub-problems leading to full exploitation of MIMD parallel computers.

Moreover the data from the channel DNS simulation seems to confirm the viability of the present algorithm to deal with complex turbulent flow configurations. At the same time it should be stressed that the capability of selecting the accuracy in determined flow regions might reveal to be a powerful tool for resolved Large Eddy Simulations in complex configurations (i.e., when approximate wall conditions are not available).

7 ACKNOWLEDGEMENTS

The present authors are grafeful for the support and the computer time provided by IRSIP (Istituto Ricerche e Sistemi Informatici Paralleli CNR, Napoli It.). We are also indebt with Dr. Di Pietro for his help and assistance when setting up the parallel version of the code. The first author likes to mention the contructive discussions and support of Prof. J. Jiménez.

REFERENCES

[1] G. De Pietro, A. Pinelli and A. Vacca, A Parallel Implementation of Spectral Multi-Domain Solver, for Incompressible Navier-Stokes Equations. In Proceedings of Parallel CFD Conference '95, CaltTech Pasadena Ca (1995), Elsevier Amsterdam.

[2] A.Chorin, A. and Marsden, A Mathematical Introduction to Fluid Mechanics, Springer-Verlag, New York, 1979.

[3] A. Pinelli and A. Vacca, Chebyshev Collocation Method and Multidomain Decomposition for the Incompressible Navier-Stokes Equations, Int. J. Num. Meth. in Fluids, 18, (1994), 781.

[4] J. Van Kan, A Second Order Accurate Pressure-Correction Scheme for Viscous Incompressible Flow, J. Sci. Stat. Comp., 7, (1986), 870.

[5] V. Agoshkov and E. Ovtchinnikov, Projection Decomposition Method, CRS4 Tech. Rep. Cagliari, Italy, 1993.

[6] A. Quarteroni, Mathematical Aspects of Domain Decomposition Methods, European Congress of Mathematics, F. Mignot (eds.), Birkhauser Boston, 1994.

[7] C. Canuto, M.Y. Hussaini, A. Quarteroni and T.A. Zang, Spectral Methods in Fluid Dynamics, Springer-Verlag, New York, 1988.

[8] E. Ovtchinnikov, On the Construction of a Well Conditioned Basis for the Projection Decomposition Method, CRS4 Tech. Rep. Cagliari, Italy, 1993.

[9] G. Golub and C. Van Loan, Matrix Computations, The Johns Hopkins University Press. Baltimore, London 1989 (II edition).

[10] J. Jimenéz and P. Moin, The Minimal Flow Unit in Near Wall Turbulence. Jou. Fluid Mech. 225, (1991), 240.

[11] T. Wei and W. Willmarth, Reynolds Number Effects on the Structure of a Turbulent Channel Flow. Jou. Fluid Mech. 204, (1989), 57.

On Improving Parallelism in the Transonic Unsteady Rotor Navier Stokes (TURNS) Code

Andrew M. Wissink[†]
Aerospace Engineering and Mechanics, University of Minnesota
107 Akerman Hall, 110 Union St. SE, Minneapolis, MN 55455

Anastasios S. Lyrintzis[‡]
School of Aeronautics and Astronautics, Purdue University
1282 Grissom Hall, West Lafayette, IN 47907

Roger C. Strawn[§]
US Army Aeroflightdynamics Directorate, Mail Stop 258-1
NASA Ames Research Center, Moffett Field, CA 94035

ABSTRACT

A parallel implementation of the three-dimensional Navier-Stokes Rotorcraft flow solver TURNS is studied. We investigate two modifications of the LU-SGS operator to improve parallel performance. The first is the Data-Parallel LU Relaxation (DP-LUR) technique. This operator uses a Jacobi sweeping procedure in place of the Gauss-Seidel sweeps in LU-SGS. The resulting algorithm is very amenable to parallel processing but requires significantly more computational work. The second approach is a Hybrid technique which maintains the nearest neighbor communication patterns of DP-LUR but uses the more efficient Gauss-Seidel sweeps of LU-SGS for the on-processor computations. The TURNS code, with the DP-LUR and Hybrid operators, is implemented on the massively parallel Thinking Machines CM-5 using a MIMD (i.e. requiring message passing) approach. The convergence qualities and the CPU time of the two implicit operators are studied for an example calculation, computing the quasi-steady three-dimensional flowfield around a helicopter blade with subsonic and transonic tip Mach numbers. Both the DP-LUR and Hybrid modifications of LU-SGS show very good parallelism, and maintain the convergence rate of LU-SGS. However, the Hybrid method uses less overall CPU time than DP-LUR.

1. INTRODUCTION

In recent years helicopters have proven to be economical and convenient vehicles with their ability to land, take-off and maneuver in areas inaccessible to fixed-wing aircraft. The ability to predict the flow around helicopter rotors is vital for the control of high-speed losses, vibration and noise.

Transonic flow is normally encountered on rotors in high-speed forward flight. Various transonic flow models have been used for the modeling of the transonic aerodynamics around the rotor. The transonic small disturbance potential formulation is the simplest approximation used. A more accurate formulation is the full potential formulation. Two examples of these full potential formulations are the FPR [1] (Full Potential Rotor) code, and the RFS2 [2] code. The main advantage of the full potential rotor codes is that they can provide a good solution at a low cost (CPU time). These codes, however, require an approximate wake model to calculate the induced downwash. The wake models are based on simple linear aerodynamics and, consequently, have a narrow range of applicability.

A more accurate CFD method is the Transonic Unsteady Rotor Navier Stokes (TURNS) code, recently developed at NASA Ames by Srinivasan and co-workers [3-5]. TURNS is capable of computing the tip vortices and the entire vortical wake as a part of the overall flowfield solution. The code has been demonstrated to calculate accurately the three-dimensional flow around the tip of a helicopter rotor in both hover and forward flight at subsonic and transonic flow conditions [3-11].

Recently, TURNS has been applied in a multidisciplinary setting, computing a near-field CFD solution that is then used as input for a Kirchhoff method that predicts the far field noise [12]. The code is currently used by NASA, the Army, various Universities, and the major US helicopter companies.

[*] Presented at the 77th AGARD F.D.P Meeting, Seville, Spain; Oct. 2-5 1995

[†] NASA Fellow, Graduate Assistant

[‡] Associate Professor

[§] Research Scientist

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

However, one drawback of TURNS is the amount of computation time it requires. An acceptable calculation with TURNS requires a supercomputer of Cray-class. A typical quasi-steady coarse-grid Euler computation by TURNS requires about 30 minutes CPU time on a Cray C-90, while an unsteady computation requires 3-4 hours. Fine-grid viscous computations require considerably more time.

Parallel computers, which include massively parallel supercomputers as well as workstation clusters, are beginning to replace traditional vector supercomputers for large scale computations due to their lower cost and high peak execution rates. At present, TURNS is inefficient on parallel machines. The main bottleneck preventing better parallel efficiency is the LU-SGS algorithm [16] used for the implicit time step. The objective of our work is to study techniques that will improve its efficiency. Thus, the majority of this paper will focus on the LU-SGS algorithm and some modifications thereof which improve its parallel efficiency. Initial results of this effort were presented in reference [13].

Although the TURNS code is primarily used for rotor CFD calculations, the solution algorithm is the same as many other CFD methods. Consequently, the parallelization procedures proposed here could readily be used for other codes that use the LU-SGS implicit operator.

2. CODE DESCRIPTION

The governing equations for the TURNS code are the unsteady, compressible, three-dimensional thin layer Navier-Stokes equations. These equations are applied in conservation form in a generalized body-conforming curvilinear coordinate system

$$\partial_\tau Q + \partial_\xi E + \partial_\eta F + \partial_\zeta G = \frac{\epsilon}{Re}\partial_\zeta S + \Re \qquad (1)$$

where $\tau = t$, $\xi = \xi(x,y,z,t)$, $\eta = \eta(x,y,z,t)$, and $\zeta = \zeta(x,y,z,t)$. The coordinate system (x,y,z,t) is attached to the blade. The vector of conserved quantities is Q, and the inviscid flux vectors E, F, and G are

$$Q = \frac{1}{J}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \qquad E = \frac{1}{J}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ UH - \xi_t p \end{bmatrix} \qquad (2)$$

$$F = \frac{1}{J}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ VH - \eta_t p \end{bmatrix} \qquad G = \frac{1}{J}\begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ WH - \zeta_t p \end{bmatrix}$$

where $H = (e + p)$ and U, V, and W, are the contravariant velocity components (e.g. $U = \xi_t + \xi_x u + \xi_y v + \xi_z w$). The cartesian velocity components u, v, and w are defined in the x, y, and z directions, respectively. The quantities ξ_t, ξ_x, ξ_y, and ξ_z are the coordinate transformation metrics and J is the Jacobian of the transformation. The pressure p is related to the conserved quantities through the perfect gas equation of state

$$p = (\gamma - 1)\left\{e - \frac{\rho}{2}(u^2 + v^2 + w^2)\right\} \qquad (3)$$

The viscous flux vector S is incorporated in the code but the calculations given in this paper are all inviscid (i.e. $\epsilon = 0$ in Eq. 1) so the viscous terms are not described here. Details can be found in [4].

The governing equations are applied to an inertial reference system that moves with the blade. Because the blade is rotating, the system is continuously unsteady. In order to get a quasi-steady starting solution, the blade must be held in in fixed position. This is done, in effect, by adding source terms to the right hand side

$$\Re = \frac{1}{J}\begin{bmatrix} 0 \\ \Omega \rho v \\ -\Omega \rho u \\ 0 \\ 0 \end{bmatrix} \qquad (4)$$

where Ω is the angular velocity of the rotor. The $\Re$ vector is used only for the quasi-steady case to get a starting solution. It is not used for unsteady calculations.

The inviscid fluxes are evaluated using Roe's upwind differencing [14] in all three directions. The use of upwinding obviates the need for user-specified artificial dissipation and improves the shock capturing in transonic flowfields. Third order accuracy is obtained using van Leer's MUSCL approach [15] and flux limiters are applied so the scheme is Total Variation Diminishing (TVD).

The final Euler discretized form of Eq. 1 in unfactored implicit delta form is

$$[I + h(\delta_\xi A^n + \delta_\eta B^n + \delta_\zeta C^n)]\Delta Q^n = -hRHS^n \qquad (5)$$

where

$$RHS^n = \partial_\xi E^n + \partial_\eta F^n + \partial_\zeta G^n - \Re^n \qquad (6)$$

I is the identity matrix, h is the time step to which the formulation is described more completely in [4], and $\Delta Q^n = Q^{n+1} - Q^n$. The 5×5 matrices A, B and C are the Jacobians of the flux vectors with respect to the conserved quantities (e.g. $A = \frac{\partial E}{\partial Q}$).

3. IMPLICIT OPERATOR

The TURNS code uses the two-factor LU-SGS (Lower-Upper Symmetric Gauss Seidel) algorithm of Yoon and Jameson [16] for the implicit time step. The LU-SGS algorithm has been used in a number of well-known CFD codes (e.g. INS3D [17], OVERFLOW [18]) primarily for it's stability properties with larger timesteps. Classic implicit methods such as Beam-Warming approximate factorization have a large factorization error (of order Δt^3) which further restricts the size of the time step. The two-factor LU-SGS method has enhanced stability along with a reduction in factorization error (order Δt^2) that make it an attractive alternative. Unfortunately, the LU-SGS method is difficult to parallelize.

The LU-SGS scheme resembles a typical LU factorization scheme with diagonal preconditioning to increase robustness. The scalar diagonal terms are obtained by use of approximate Jacobians, avoiding costly matrix inversions. The Jacobian terms A,B, and C in Eq. 5 are split into "+" and "−" parts, with positive parts constituting only the positive eigenvalues and negative parts constituting only the negative eigenvalues. The positive matrix is backward differenced and the negative matrix is forward differenced, as follows

$$\begin{aligned} A_\xi &= \delta_\xi^- A^+ + \delta_\xi^+ A^- \\ &= A_j^+ - A_{j-1}^+ + A_{j+1}^- - A_j^- \end{aligned} \tag{7}$$

This splitting ensures diagonal dominance. Approximate Jacobians are constructed using a spectral approximation

$$A^\pm = s_A^\pm \frac{1}{2}(A \pm \rho_A I) \pm \varepsilon \rho_A I \tag{8}$$

where ρ_A is the spectral radius of A (in the ξ direction).

$$\rho_A = max\left[|\lambda_A|\right] = |U| + a\,|\nabla\xi| \tag{9}$$

ε is some small value (e.g. .001), and $s_A^\pm$ is defined as

$$s_A^\pm = \begin{cases} 1 & if \pm (U \pm a\,|\nabla\xi|) \geq 0 \\ 0 & otherwise \end{cases} \tag{10}$$

The same procedure is used in the η and ζ directions to form the B and C terms.

Substituting this development into Eq. 5, we arrive at a system of the form

$$LD^{-1}U\Delta Q^n = -hRHS^n \tag{11}$$

where

$$\begin{aligned} D &= I + h\,(\rho_A + \rho_B + \rho_C)_{j,k,l} \\ L &= D - h\,(A_{j-1}^+ + B_{k-1}^+ + C_{l-1}^+) \\ U &= D + h\,(A_{j+1}^- + B_{k+1}^- + C_{l+1}^-) \end{aligned} \tag{12}$$

D is a diagonal matrix, and the two step LU decomposition can be performed by

$$\begin{aligned} L\Delta Q^* &= -hRHS^n \\ U\Delta Q^n &= D\Delta Q^* \end{aligned} \tag{13}$$

which can also be written

$$\begin{aligned} \delta Q_{j,k,l}^* &= D^{-1}h \begin{bmatrix} -RHS^n + (A_{j-1}^+ \delta Q_{j-1}^* \\ +B_{k-1}^+ \delta Q_{k-1}^* + C_{l-1}^+ \delta Q_{l-1}^*) \end{bmatrix} \\ \delta Q_{j,k,l}^n &= \delta Q_{j,k,l}^* - D^{-1}h \begin{bmatrix} A_{j+1}^- \delta Q_{j+1}^* + B_{k+1}^- \\ \delta Q_{k+1}^* + C_{l+1}^- \delta Q_{l+1}^* \end{bmatrix} \end{aligned} \tag{14}$$

In the first step of (14), sweeps updating δQ^* are performed in the positive direction (that is, from 1 to $j_{max}, k_{max}, l_{max}$) through the solution domain. The second step then computes δQ^n by sweeping back through the domain in the opposite direction. This algorithm can be vectorized using a hyperplane approach, as outlined in [19]. Vectorization is done across hyperplanes in which *j+k+l=const.* This is outlined in Fig. 1.

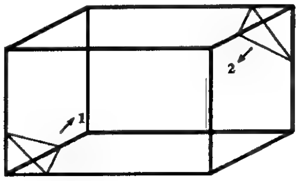


Figure 1: Domain sweeping strategy used by LU-SGS algorithm. Can vectorize on hyperplanes where *j+k+l = const.*

While the hyperplane approach leads to good vector execution rates, it is difficult to parallelize for two reasons; 1) the size of the hyperplanes vary throughout the grid, leading to load balancing problems, and 2) there is a recursion between the planes, leading to a large amount of communication.

Parallelization of the LU-SGS algorithm in (14) has been addressed by other researchers. Barszcz et al. [19] implemented the LU-SSOR algorithm, which is similar to LU-SGS, on a massively parallel machine by restructuring the data-layout using a skew-hyperplane approach. Although they were able to extract reasonable parallelism with this approach the data-layout is complex and considerable effort was required to implement the domain partitioning in an efficient manner when using a MIMD (Multiple Instruction Multiple Data) implementation. Also, the restructuring of data on the left hand side in turn causes the right hand side layout to be skewed and extra communication is required. Overall, the LU-SGS algorithm (14) is not conducive to efficient parallel execution.

Several researchers have proposed modifications of the LU-SGS algorithm to make it more parallelizable. Candler et al. [21, 22] have investigated a modification called Data-Parallel LU Relaxation (DP-LUR), which has shown excellent results in a data-parallel environment. It is used in this study and is discussed more thoroughly in section 3.1. Wong et al. [20] have investigated a domain decomposition implementation of LU-SGS. For two-dimensional steady state reacting flow problems, they found that, while the convergence rate of the operator is reduced with the domain breakup, the affect is relatively weak (e.g. with 64 subdomains, the number of iterations increases by less than 20%). Thus, the domain decomposition strategy appears promising, and is used as a basis for the Hybrid algorithm, discussed in section 3.2.

3.1 DP-LUR Method

A modification of LU-SGS, referred to as Data-Parallel LU Relaxation, has been introduced by Candler et al. [21, 22] for solving hypersonic flow problems. Essentially, the modification involves transferring the nondiagonal terms to the right hand side and using values from the previous iteration for these terms. The modified operator then becomes Jacobi-like and requires only nearest neighbor communication. This operator has been found to be very efficient in a data-parallel environment (e.g. [22, 23]). The DP-LUR modification of the LU-SGS algorithm is given in (15).

```
$\delta Q^0_{j,k,l} = D^{-1} \cdot hRHS^n$

For $i = 1, \ldots, i_{max}$ Do
    $\delta Q^{(i)}_{j,k,l} = D^{-1} \cdot h \begin{bmatrix} RHS^n + \\ A^+_{j-1}\delta Q^{(i-1)}_{j-1} - A^-_{j+1}\delta Q^{(i-1)}_{j+1} + \\ B^+_{k-1}\delta Q^{(i-1)}_{k-1} - B^-_{k+1}\delta Q^{(i-1)}_{k+1} + \\ C^+_{l-1}\delta Q^{(i-1)}_{l-1} - C^-_{l+1}\delta Q^{(i-1)}_{l+1} \end{bmatrix}$        (15)
End Do
    $\delta Q^n_{j,k,l} = \delta Q^{(i_{max})}_{j,k,l}$
```

The main difference between the LU-SGS and DP-LUR algorithms is that a Jacobi sweeping strategy is used in DP-LUR while Gauss-Seidel sweeps are used in LU-SGS. The advantage of using Jacobi sweeps is that there is no recursion of data and only nearest neighbor communication is required at each node. Thus, it can be completely load balanced with communications only at the borders of each partition (Fig. 2).

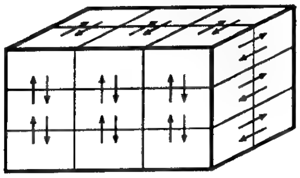


Figure 2: Jacobi Sweeping Strategy of DP-LUR algorithm. Load balanced parallelism with nearest neighbor communication.

Although DP-LUR is more amenable to parallel processing than LU-SGS, the use of Jacobi sweeps leads to a larger amount of computational work. It is well-known that a Jacobi method will have a theoretically slower convergence rate than Gauss-Seidel. Multiple sweeps (e.g. 4-6) are therefore required inside Eq. (15) to maintain a comparable convergence rate to LU-SGS. Although DP-LUR can be executed efficiently on a parallel machine, the added computational cost is a significant penalty, the specifics of which are discussed in section 5.1. The question is whether the computational penalty of DP-LUR is the best that we can do.

3.2 Hybrid Method

The motivation behind development of the Hybrid approach is to replace a source of inefficiency in DP-LUR. The DP-LUR algorithm was developed primarily for data-parallel computations. Its convergence is independent of the number of processors used because the same Jacobi sweeping strategy that allows nearest neighbor communications between the processors is also used for the computations on each processor. Doing the on-processor computations with Jacobi sweeps is a source of inefficiency, since the computational work can be performed more efficiently with the Gauss-Seidel sweeps of LU-SGS. The strategy behind the Hybrid approach is to use the communications structures of the DP-LUR algorithm, to maintain load-balanced parallelism with nearest neighbor communications, along with the more efficient LU-SGS algorithm for the on-processor computations. The algorithm is referred to as the Hybrid approach because it retains features of both the LU-SGS and DP-LUR algorithms.

```
$\delta Q^{(0)}_{j,k,l} = D^{-1} \cdot hRHS^n$

For $i = 1, \ldots, i_{max}$ Do
$\delta Q^{*(i)}_{j,k,l} = \delta Q^{(i-1)}_{j,k,l}$        (16)
$\delta Q^{*(i)}_{j,k,l} = D^{-1} \cdot h \begin{bmatrix} -RHS^n + (A^+_{j-1}\delta Q^{*(i)}_{j-1} \\ +B^+_{k-1}\delta Q^{*(i)}_{k-1} + C^+_{l-1}\delta Q^{*(i)}_{l-1}) \end{bmatrix}$
$\delta Q^{(i)}_{j,k,l} = \delta Q^{*(i)}_{j,k,l} - D^{-1} \cdot h \begin{bmatrix} A^-_{j+1}\delta Q^{*(i)}_{j+1} + B^-_{k+1} \\ \delta Q^{*(i)}_{k+1} + C^-_{l+1}\delta Q^{*(i)}_{l+1} \end{bmatrix}$
End Do

$\delta Q^n_{j,k,l} = \delta Q^{(i_{max})}_{j,k,l}$
```

The equations used inside the sweeps of (16) are the same as those used by the LU-SGS algorithm (14). Thus, with 1 sweep (i.e. $i_{max} = 1$), the Hybrid algorithm is very similar to a domain decomposition implementation of LU-SGS, the only difference being the initial condition on the first line of (16). The use of multiple sweeps improves the convergence rate, making up for the loss of connection in the domain decomposition.

On 1 processor (with 1 sweep), the method is identical to the original LU-SGS algorithm. On many processors, (i.e. in the limiting condition where the number of processors approaches the number of gridpoints) the Hybrid method is identical to the DP-LUR algorithm. The computational workload of the Hybrid algorithm, therefore, is dependent upon the number of processors used. The algorithm should be most efficient with few processors, and should always require less computational work than DP-LUR.

Parallel implementation of the Hybrid algorithm is done in essentially the same way as DP-LUR. Border data is communicated to nearest neighbors at the beginning of each sweep and each processor performs the standard LU-SGS algorithm on its domain. Because the size of the domains corresponds with the number of processors used, the convergence will be different with different processor partitions. However, like DP-LUR, the Hybrid algorithm maintains load balanced parallelism with only nearest neighbor communications.

4. PARALLEL IMPLEMENTATION

A MIMD approach (i.e. requiring message passing) is used for parallel implementation. There are two reasons for choosing the MIMD approach over a SIMD (Single Instruction Multiple Data) or data-parallel approach; 1) Code portability; because message passing codes are more portable to different parallel architectures (e.g. from massively parallel supercomputers to workstation clusters), and 2) Ease of implementation; since the original code is over 6000 lines, it is much easier to add message passing directives to the existing code than rewrite the entire code in a High Performance Fortran type language (e.g. CMFortran). To ensure easy portability of the code, a set of generic message passing subroutines was used. With this protocol, the specific message passing commands can be altered in one line of the code rather than throughout, making conversion to different message passing languages, such as PVM (Parallel Virtual Machine) and MPI (Message Passing Interface), a relatively short procedure.

Fig. 3 shows the breakup of the three-dimensional solution domain. The flowfield domain is layed out on a two-dimensional array of processors. The flowfield is split in the wraparound (J) and spanwise (K) directions. The normal direction (L) is left intact so that the implementation of surface boundary conditions is unchanged from the existing serial code. A single layer of ghost cells is placed on the border of each processor, providing a location where the communicated data can be stored.

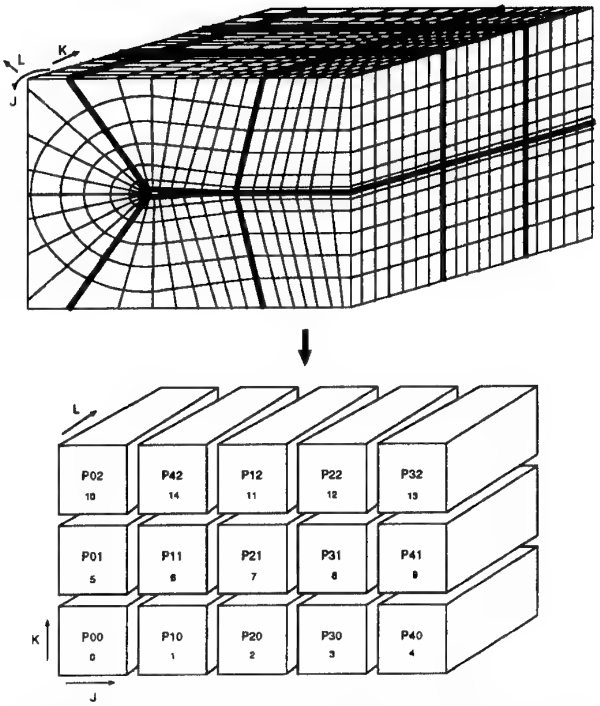


Figure 3: Partitioning the three-dimensional domain on a two-dimensional array of processors.

The communications between neighboring processors is done once during each of the inner sweeps of the DP-LUR and Hybrid algorithms, totaling 4 × i_{max} communication steps. One communication step is required to pass information to form the RHS, since third order accuracy requires data at the $j, k, l \pm 2$ points. This communication step could be eliminated if a layer of two ghost cells were used but this increases memory and communications. The boundary conditions at the flowfield borders and on the rotor blade can be imposed locally on each processor, but communication is required in the wake region where $L = 1$ to enforce the boundary condition where the C-H grid collapses and data is averaged across this wake plane. Only the processors holding this data perform communications and only one communication step is needed.

5. RESULTS AND DISCUSSION

The TURNS code with the DP-LUR and Hybrid implicit operators have been implemented on the massively parallel Thinking Machines CM-5 at the Army High Performance Computing Research Center (AHPCRC) in Minneapolis, MN. The CM-5 has a total of 896 processors, configurable in processor partitions of 64, 256, and 512 processors. The implementation is performed by adding message passing calls to the existing Fortran 77 code. The message passing calls are taken from the CMMD library, which is supported by Thinking Machines, Inc.

Each processor on the CM-5 has a peak performance of 5 Mflops/Processor. Vector Units (VU's) exist on each processor that increase the performance substantially (e.g. from 5 Mflops/processor to 128 Mflops/processor). Unfortunately, the only way to utilize the VU's at this time is to rewrite the code in CMFortran, a High-Performance-Fortran type language. Since TURNS is over 6K lines, rewriting the code would require considerable effort and was one of the main reasons we chose the MIMD implementation in the first place. In addition, rewriting the code to CMFortan would eliminate code portability. Consequently, the results presented here are determined without utilizing the VU's. Although this degrades the performance on the CM-5, it is not a big drawback overall, because our future plans are to run the code on parallel systems such as the IBM SP-2 and workstation clusters, which do not have vector units.

The code is run for a test problem that computes the quasi-steady flowfield around a symmetric OLS blade. The OLS blade has a sectional airfoil thickness to chord ratio of 9.71% and is a 1/7 scale model of the main rotor for the Army's AH-1 helicopter. A $135 \times 50 \times 35$ C-H type grid is used, with the domain extending eight chords in all directions. The upper half of the grid is shown in Fig. 4. We chose to use this particular grid and airfoil because they were used for calculations in the aeroacoustic study in [12]. Unfortunately, the unusual mesh dimensions cannot be partitioned in a way that exactly matches the processor partitions on the CM-5 (64, 256, and 512 processors). For example, the J dimension has only odd factors so it is impossible to partition it on an even number of processors. We did break up the mesh in a way that used *most* of the processors in the partition. For the 64 node partition, the mesh was broken in 19 points in the J direction, and 3 points in the K direction, giving a total of 57 processors. For the 256 node partition, the mesh was broken in 19 points and 12 points in the J and K directions, respectively, giving 228 processors. Finally, for the 512 node partition, the mesh was broken in 19 points and 24 points in the J and K directions, giving 456 processors. When executing the code, the remaining processors in the partition sit idle. Generally, most newer machines (e.g. IBM SP-2) allow the user to choose the exact number of processors they want for their partition, so this will most likely not be an issue on more modern machines.

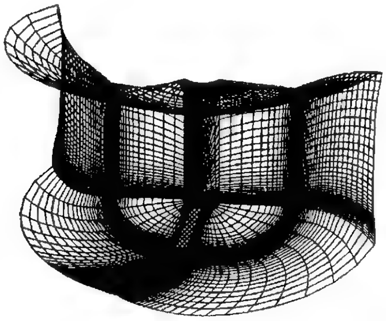


Figure 4: Upper half of the $135 \times 50 \times 35$ C-H type grid used for OLS airfoil calculations on the CM-5.

The three dimensional quasi-steady starting solution is computed around the rotating blade in subsonic conditions, with $M_{tip} = 0.664$, and a more transonic condition, with $M_{tip} = 0.80$. In both cases, the freestream Mach number is $M_\infty = 0.17$ and the blade position is fixed at zero degrees azimuth angle (Fig. 5). It should be noted that the first case, $M_{tip} = 0.664$, is a realistic test case for rotor calculations. The $M_{tip} = 0.800$ case, however, is far too transonic to be used in a practical helicopter application. It was added as an extreme test case to investigate the behavior of the implicit solvers with more nonlinear transonic flows.

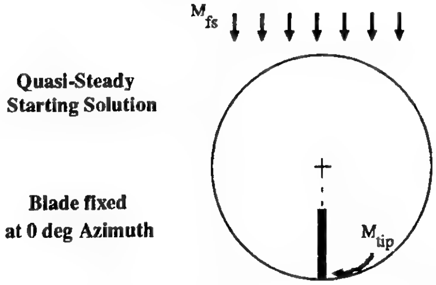


Figure 5: Quasi-Steady solution. Blade fixed at zero degrees azimuth angle.

It should be also be noted that results are presented for a quasi-steady fixed blade case instead of an unsteady case because the convergence behavior of the implicit solvers can be quantified most easily with this quasi-steady case. It is difficult to investigate convergence behavior with an unsteady case without also verifying time-accuracy for the implicit solver. This does not indicate that the method is unable perform unsteady runs. The same algorithm is used for time-accurate unsteady cases so these cases can be run without further modifications to the algorithm.

5.1 DP-LUR Results

The results of timings of TURNS with the DP-LUR algorithm on 57, 228, and 456 processors are given in Tables 1 and 2, for the $M_{tip} = 0.664$ and $M_{tip} = 0.800$ cases, respectively. The method is stopped when the density residual drops by two orders of magnitude below its maximum value. Plots of the L2-norm density residual vs. number of iterations are shown for the two cases in Figs. 6 and 7. The convergence of the original LU-SGS method, run on a single processor, is also shown in the plots for comparison purposes.

Table 1 - Timing Results on the CM-5 for TURNS with DP-LUR for subsonic test case. 135 × 50 × 35 mesh, $M_{tip} = 0.664$, density residual converged to 5×10^{-7}.

Procs	Iterations	% Comm.	Tot. Time
5 sweeps			
57	436	10.4 %	9330 sec
228	440	15.3 %	2508 sec
456	438	21.0 %	1445 sec
6 sweeps			
57	351	9.2 %	8505 sec
228	350	15.1 %	2233 sec
456	353	19.9 %	1292 sec
7 sweeps			
57	304	9.6 %	8229 sec
228	304	16.6 %	2110 sec
456	306	20.6 %	1224 sec

Table 2 - Timing Results on the CM-5 for TURNS with DP-LUR for transonic test case. 135 × 50 × 35 mesh, $M_{tip} = 0.800$, density residual converged to 5×10^{-7}.

Procs	Iterations	% Comm.	Tot. Time
5 sweeps			
57	464	10.0 %	9902 sec
228	457	14.8 %	2628 sec
456	465	20.8 %	1511 sec
6 sweeps			
57	379	10.0 %	9210 sec
228	380	15.1 %	2424 sec
456	383	19.9 %	1402 sec
7 sweeps			
57	335	9.6 %	9068 sec
228	335	16.6 %	2383 sec
456	345	20.6 %	1380 sec

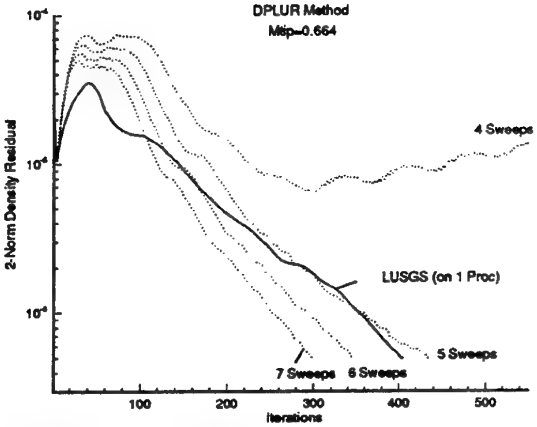


Figure 6: Convergence of TURNS with DP-LUR method. $M_{tip} = 0.664$

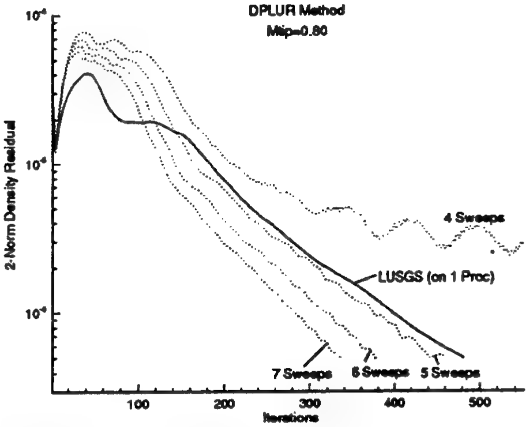


Figure 7: Convergence of TURNS with DP-LUR method. $M_{tip} = 0.80$

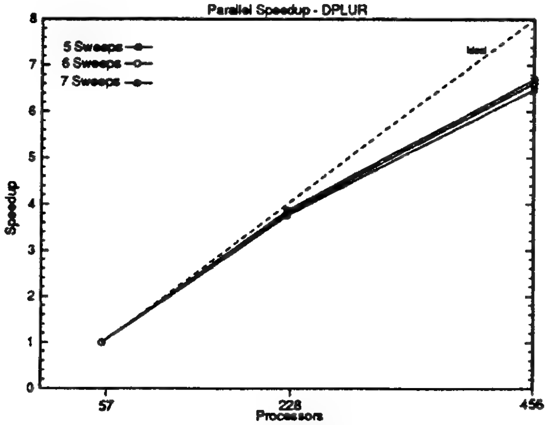


Figure 8: Parallel Speedups of the time per iteration using the DP-LUR operator

Table 3 - Timing Results on the CM-5 for TURNS with the Hybrid method for subsonic test case. $135 \times 50 \times 35$ mesh, $M_{tip} = 0.664$, density residual converged to 5×10^{-7}.

Procs	Iterations	% Comm.	Tot. Time
1 sweep			
57	461	10.3 %	4937 sec
228	470	15.1 %	1434 sec
456	502	18.8 %	863 sec
2 sweeps			
57	394	10.1 %	5410 sec
228	398	14.8 %	1524 sec
456	404	20.6 %	889 sec
3 sweeps			
57	386	10.0 %	6423 sec
228	385	14.8 %	1771 sec
456	385	19.7 %	1012 sec

Table 4 - Timing Results on the CM-5 for TURNS with the Hybrid method for transonic test case. $135 \times 50 \times 35$ mesh, $M_{tip} = 0.800$, density residual converged to 5×10^{-7}.

Procs	Iterations	% Comm.	Tot. Time
1 sweep			
57	531	10.6 %	5719 sec
228	558	15.3 %	1707 sec
456	580	18.8 %	998 sec
2 sweeps			
57	483	10.1 %	6568 sec
228	485	14.8 %	1858 sec
456	492	20.4 %	1082 sec
3 sweeps			
57	467	9.9 %	7748 sec
228	466	14.4 %	2143 sec
456	470	19.2 %	1226 sec

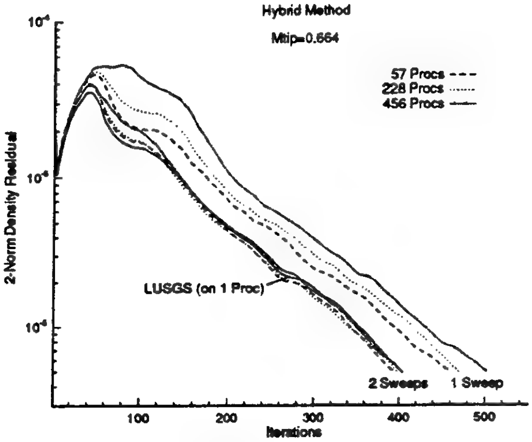


Figure 9: Convergence of TURNS with Hybrid method. $M_{tip} = 0.664$

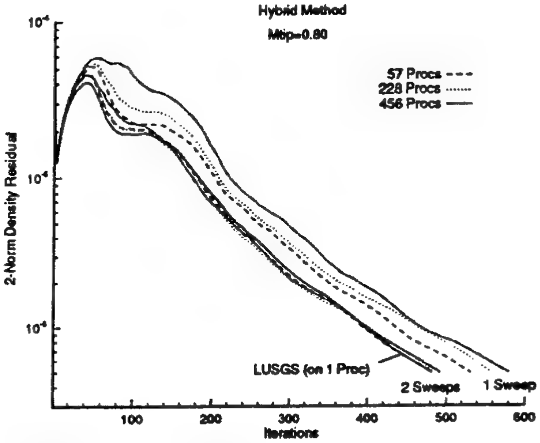


Figure 10: Convergence of TURNS with Hybrid method. $M_{tip} = 0.80$

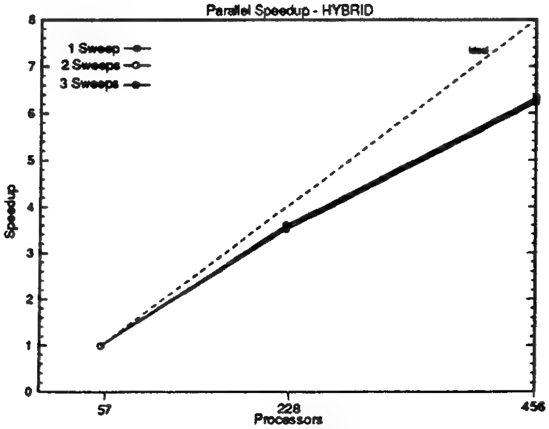


Figure 11: Parallel Speedups of the time per iteration of TURNS using the Hybrid operator

The convergence plots show that a minimum of 5 inner sweeps (i.e. $i_{max} = 5$) of DP-LUR are required to converge the solution. In both the subsonic and transonic cases, 4 sweeps began to diverge. For the $M_{tip} = 0.664$ case, 5 sweeps gives slightly worse convergence than single processor LU-SGS while 6 sweeps gives slightly better. For the $M_{tip} = 0.800$ case, 5 sweeps of DP-LUR gives about the same convergence as single processor LU-SGS, and 6 sweeps is better. This seems to indicate that DP-LUR maintains a good level of robustness for transonic cases, since it requires less inner sweeps to maintain the convergence rate of LU-SGS. The single processor LU-SGS method requires the work of approximately 1.8 sweeps of DP-LUR. Consequently, these results show that, in order to maintain the same convergence rate, the DP-LUR implicit operator requires about 3 times the computational work of single processor LU-SGS.

Timings of the DP-LUR method indicate that more sweeps seems to be the better choice. The overall CPU time with 7 sweeps is fastest, but the difference between 6 and 7 sweeps is small (less than 2%). Each additional sweep increases the CPU time per iteration by 10-15%. Communication represents a relatively small percentage of the total CPU time. The communication percentage increases with increasing number of processors. Also, the percentages tend to fluctuate for different cases which is probably due to the fact that these runs were done on a loaded rather than dedicated machine.

It should be noted that, in theory, the solution using DP-LUR is the same regardless of the number of processors used, so the number of iterations should be the same for all processor partitions. However, Tables 1 and 2 show that the implementation did show some slight discrepancies in the number of iterations. Generally, the differences are small (less than 4%) and we attribute them to numerical roundoff in the machine. Differences in the overall solution are indistinguishable for the different partition sizes.

A plot of the parallel speedups of the time per iteration of TURNS with DP-LUR is shown in Fig. 8. The speedup from 57 to 228 processors is nearly linear, but some falloff is noted for 456 processors. This is believed to be due to the relatively small problem size of 236,250 gridpoints. It is expected that the speedup will be more linear with larger problems. The parallel speedup increases slightly for a larger number of sweeps, since the amount of computational work goes up. However, the difference is not significant.

5.2 Hybrid Results

Results of timings with the Hybrid algorithm are given in Tables 3 and 4, for the $M_{tip} = 0.664$ and $M_{tip} = 0.800$ cases, respectively. Plots of the density residual vs. CPU time are given in Figs. 9 and 10.

The efficiency of the Hybrid method is apparent in the number of inner sweeps required for convergence. While DP-LUR required a minimum of 5 sweeps, the Hybrid method converges at a comparable rate to single processor LU-SGS with only 1 sweep. This is due to the more efficient Gauss-Seidel procedure used for the on-processor computations. With 2 sweeps, the convergence of the Hybrid method is almost identical to single processor LU-SGS. With one sweep, there is significant spread between the convergence curves for the different numbers of processors, but with 2 sweeps, the spread is reduced considerably so that all processor partitions follow essentially the same convergence path as LU-SGS. Although it is not shown in the figures, the convergence plot with 3 sweeps is only slightly better than with 2, and it is therefore not plotted to avoid the graph from becoming too crowded.

The Hybrid method is considerably faster than DP-LUR. The CPU times of the Hybrid method are only 55-60% those of DP-LUR. This is due to the larger amount of computational work in DP-LUR, because a larger number of sweeps are required for convergence.

It should be pointed out that each sweep with DP-LUR involves only a single sweep through the domain on each processor, whereas the Hybrid method performs the two-step LU-SGS algorithm on each processor, performing two sweeps through the domain. Thus, each sweep of the Hybrid method is approximately equivalent to the work of two sweeps in DP-LUR. This is indicated in the CPU times; the CPU time using 6 sweeps of DP-LUR is approximately equal to 3 sweeps using the Hybrid method.

Using 1 sweep in the Hybrid method gives the best CPU time, but requires 17-18% more iterations than single processor LU-SGS. The CPU time with 2 sweeps is worse than that of 1 sweep by about 8%, but the convergence rate is much closer to that of single processor LU-SGS. When 3 sweeps are used, the convergence is only slightly better (a reduction in iterations of less than 5%) than 2 sweeps, while the CPU time is about 11-15% more. Thus, 3 sweeps or more appears to be unnecessary.

A plot of the parallel speedups of the time per iteration is shown in Fig. 11. The parallel speedups are essentially the same as with DP-LUR.

6. SUMMARY AND CONCLUSIONS

A strategy is presented for implementing the three-dimensional Navier-Stokes Rotorcraft CFD code TURNS on massively parallel computer architectures. The main portion of the code that is difficult to parallelize is the implicit timestep using the LU-SGS operator. We study two modifications of this operator that make it more amenable to parallel implementation. The first is the Data-Parallel LU Relaxation (DP-LUR) technique, which essentially replaces the Gauss-Seidel sweeps in LU-SGS with Jacobi sweeps, and uses multiple sweeps of the domain to maintain the same convergence rate. The second is a new approach that couples the Jacobi communication strategy of DP-LUR with Gauss-Seidel sweeps of LU-SGS for the on-processor computations. It also uses multiple inner sweeps to maintain the convergence rate of LU-SGS. Because this second approach retains features of both the DP-LUR and LU-SGS algorithms, we call it a Hybrid method.

The TURNS code is tested on the Thinking Machines CM-5, using a MIMD approach for parallel implementation. It is run for an Euler quasi-steady calculation with 236,250 gridpoints, computing the flow around the tip of a helicopter blade rotating with subsonic and transonic tip Mach numbers. Results from various processor partitions show that both the DP-LUR and Hybrid modifications of LU-SGS are very parallelizable, showing good parallel speedups. Both methods are also able to maintain the convergence qualities of original LU-SGS for all test cases. The Hybrid method, however, requires less CPU time due to lower computational work requirements. The DP-LUR modification of LU-SGS causes the amount of computational work in the implicit solver to increase threefold, to maintain the same convergence rate. The Hybrid modification, however, can match to within 25% the convergence rate of single processor LU-SGS with no increase in the computational work. It can exactly match the convergence rate with twice as much work in the implicit solver, yielding CPU times that are only 8% higher than the single sweep cases. Overall, the CPU times for the Hybrid method are only 55-60% those of DP-LUR.

The computational work required of the Hybrid approach on a parallel machine will always be less than that of DP-LUR. On a few processors, the amount of computational work will be about the same as LU-SGS. The Hybrid approach is therefore ideally suited for machines that have smaller numbers of more powerful, non-vectorized, processors. One example of a machine that fits this category is the 150 processor IBM SP-2. We are currently implementing the code on the IBM SP-2 at NASA Ames, and expect better CPU times than what were obtained on the CM-5.

Finally, although the TURNS code is used primarily for rotorcraft CFD applications, the parallelization strategy is not unique to this application. The parallelization procedures proposed here could be readily used for other CFD codes that use the LU-SGS algorithm.

ACKNOWLEDGMENTS

The first author was supported by a NASA Graduate Student Fellowship. This work was supported by allocation grants from the Minnesota Supercomputer Institute (MSI) and Cray Research, Inc. The work is also supported in part by grant NSF CCR-9496327 and by the Army Research Office contract number DAAL03- 89-C-0038 with the University of Minnesota Army High Performance Computing Research Center (AHPCRC) and the DOD Shared Resource Center at the AHPCRC.

References

[1] Strawn, R. C., and Caradonna, F. X., "Conservative Full Potential Model for Unsteady Transonic Rotor Flows," *AIAA Journal*, Vol. 25, No. 2, Feb. 1987, pp. 193-198.

[2] Bridgeman, J. O., Steger, J. L., and Caradonna, F. X., "A Conservative Finite-Difference Algorithm for the Unsteady Transonic Potential Equation in Generalized Coordinates," AIAA Paper 82-1388, 9th Atmospheric Flight Mechanics Conference, San Diego, CA, Aug. 1982.

[3] Srinivasan, G. R., "A Free-Wake Euler and Navier-Stokes CFD Method and its Application to Helicopter Rotors Including Dynamic Stall," JAI Associates, Inc., Technical Report 93-01, November 1993.

[4] Srinivasan, G. R., Baeder, J. D., Obayashi, S., and McCroskey, W. J., "Flowfield of a Lifting Rotor in Hover: A Navier Stokes Simulation," *AIAA Journal*, Vol. 30, No. 10, Oct. 1992, pp. 2371-2378.

[5] Srinivasan, G. R., and Baeder, J. D., "TURNS: A Free-Wake Euler/Navier-Stokes Numerical Method for Helicopter Rotors," *AIAA Journal*, Vol. 31, No. 5, May 1993, pp. 959-962.

[6] Srinivasan, G.R., and Raghavan, V., Duque, E. P. N., and McCroskey, W., J., "Flowfield of a Lifting Rotor in Hover - A Navier Stokes Simulation," *AIAA Journal*, Vol. 30, No. 10, October 1992.

[7] Srinivasan, G.R., and Ahmad, J.U., "Navier Stokes Simulation of Rotor-Body Flowfields in hover Using Overset Grids," Proceedings of the Nineteenth European Rotorcraft Forum, Paper No. C15, September 1993, Cernobbio Italy.

[8] Duque, E.P.N., and Srinivasan, G.R., "Numerical Simulation of a Hovering Rotor Using Embedded Grids," Proceedings of the 48th Annual Forum of the American Helicopter Society, Washington DC, June 1992.

[9] Duque, E.P.N., "A Structured/Unstructured Embedded Grid Solver for Helicopter Rotor Flows," Proceedings fo the 50th Annual Forum of the American Helicopter Society, Vol. II, Washington DC, May 1994, pp. 1249-1257.

[10] Baeder, J.D., Gallman, J.M., and Yu, Y.H., "A Computational Study of the Aeroacoustics of Rotors in Hover," Proceedings of 49th Annual Forum of the American Helicopter Society, St. Louis, Missouri, May 1993, pp. 55-71.

[11] Baeder, J.D., and Srinivasan, G.R., "Computational Aeroacoustic Study of Isolated Blade Vortex Interaction Noise," AHS Specialists' Aeromechanics Conference, San Francisco, CA, Jan 1994.

[12] Strawn, R. C., Biswas, R., and Lyrintzis, A. S., "Helicopter Noise Predictions using Kirchhoff Methods," presented at the 51st Annual Forum of the American Helicopter Society, 9–11, May, 1995, Fort Worth, Texas; also, to be published in the Journal of Computational Acoustics.

[13] Wissink, A.M., Lyrintzis, A.S., and Strawn, R.C. "On the Parallelization of the Transonic Unsteady Rotor Navier Stokes Code," presented at the Computational Aerosciences Conference, March 11-13, 1995, Santa Clara, CA.

[14] Roe, P.L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 3, 1981, pp. 357-372.

[15] Anderson, W.K., Thomas, J.L., and van Leer, B., "A Comparison of Finite Volume Flux Vector Splittings for the Euler Equations," AIAA Paper 85-0122, Jan. 1985.

[16] Yoon, S., and Jameson, A., "A Lower-Upper Symmetric Gauss Seidel Method for the Euler and Navier Stokes Equations," *AIAA J.*, Vol. 26, 1988, pp. 1025-1026.

[17] Yoon, S., and Kwak, D., "Three-Dimensional Incompressible Navier-Stokes Solver Using Lower-Upper Symmetric-Gauss-Seidel Algorithm," *AIAA J.*, Vol. 29, No. 6, June 1991, pp. 874-875.

[18] Kandula, M., and Buning, P.G., "Implementation of LU-SGS Algorithm and Roe Upwinding Scheme in OVERFLOW Thin-Layer Navier-Stokes Code," AIAA Paper 94-2357, June 1994.

[19] Barszcz, E., Fatoohi, R., Venkatakrishnan, V., and Weeratunga, S., "Solution of Regular, Sparse Triangular Linear Systems on Vector and Distributed-Memory Multiprocessors," NASA Report RNR-93-007, April 1993.

[20] Wong, C.C., Blottner, F.G., and Payne, J.L., "A Domain Decomposition Study of Massively Parallel Computing in Compressible Gas Dynamics," AIAA Paper 95-0572, Jan. 1995.

[21] Candler, G.V., Olynick, D.R., "Hypersonic Flow Simulations Using a Diagonal Implicit Method," presented at the 10th International Conference on Computing Methods in Applied Sciences and Engineering, Paris France, Feb. 1992.

[22] Candler, G.V., Wright, M., and McDonald, J.D., "A Data Parallel LU-SGS Method for Reacting Flows," *AIAA Journal*, Vol. 32, No. 12, Dec. 1994, pp. 2380-2386.

[23] Wright, M.J., Candler, G.V., and Prampolini, M. "A Data Parallel LU Relaxation Method for the Navier Stokes Equations," AIAA Paper 95-1750, 1995.

DEVELOPMENT OF A PARALLEL IMPLICIT ALGORITHM FOR CFD CALCULATIONS

F. Dias d'Almeida, F.A. Castro, J.M.L.M. Palma and P. Vasconcelos
Faculdade de Engenharia da Universidade do Porto
Rua dos Bragas, 4099 Porto CODEX, Portugal

Abstract

The present article reports on further developments of an implicit coupled algorithm for fluid flow equations. Mass and momentum conservation equations are solved as part of one large system of equations in one single step. Iterations are needed because of nonlinearities only. The algorithm requires no under-relaxation factors and can reach convergence in a reduced number of iterations, compared to decoupled approaches. This article describes improvements leading to reduction of both memory and computing time. The algorithm exceeds the memory requirements of the SIMPLE algorithm of Patankar and Spalding by a factor of K^2, where K is the number of independent variables. Computing time reduction was achieved by using GMRES and a preconditioner based on incomplete LU factorization. The algorithm compares favourably with conventional decoupled approaches. To overcome the high memory requirements and enable the simulation of large physical problems two different approaches for parallelization were also tested, at the expense of increased computing time.

1 INTRODUCTION

The SIMPLE [1, 2] algorithm is amongst the most widely used algorithms for solving the fluid flow equations. The difficulties of convergence of SIMPLE when dealing with large problems, either in terms of physical complexity or grid size, are well known and have been discussed in the open literature (e.g.: [3] [4], [5]). SIMPLE is relatively easy to implement and accommodate for increased number of transport equations, but its sensitivity to numerical aspects as for instance, under-relaxation factors [6] has led to many research efforts and even new algorithms (e.g.: [7]), sometimes closely related to SIMPLE (e.g: [8] [9]).

The algorithm discussed in this article (designated DIRECTO [10]) solves the fluid flow equations as a complete coupled system. The cell face velocities are predicted using a momentum equation, which once replaced into the continuity equation leads to a pressure equation fully coupled to the velocity field. No simplification is made at this stage, the equation is exact, opposed to pressure (or pressure-correction) equation derived from segregated [11] algorithms.

The code development was made using the two classical geometries of a two-dimensional cavity with a sliding lid and a backward-facing step. Results [12] show that for instance, in case of the square cavity with sliding lid at Re=1000 the DIRECTO algorithm with LU factorization converged in 8 iterations, independently of grid size, for a residual error of 1×10^{-3}. On the other hand the SIMPLE algorithm although requiring 186 iterations for a grid of 64×64, used 10× less CPU time. This order of magnitude ratio was reduced by using *Block Band LU* factorization [13] and GMRES [14], since at each iteration there is the solution of a large, sparse, unsymmetric, block-band (block tridiagonal) linear system.

The need for finer grids, mainly on complex geometries, leads to very large systems of equations requiring the use of secondary storage and large CPU times. Parallel architectures with distributed memory may be one answer to those problems. The main drawbacks are the communication between processors and additional computations. In this work a cluster of 4 DEC AlphaStation AXP, models 500S and 600S, connected by FDDI (Fiber Distributed Data Interface) and Gigaswitch using PVM (Parallel Virtual Machine) [15, 16] was used as a parallel environment. PVM is a software package that allows the concurrent use of heterogeneous processing elements.

The article is made up of 3 major Sections. In Section 2 the algorithm is described. Section 3 discusses the results with respect to accuracy, memory requirements and computing time, including a discussion on the linear solvers and parallelization. Section 4 concludes the article.

2 MATHEMATICAL MODEL

The governing equations for two-dimensional incompressible Newtonian and isothermal flows are, in tensorial notation,

$$\frac{\partial U_i}{\partial x_i} = 0\,, \tag{1}$$

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

and,

$$\rho\frac{\partial\left(U_iU_j\right)}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \mu\frac{\partial}{\partial x_j}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) , \tag{2}$$

where U_i is the velocity component along the x_i direction and P, ρ and μ are the static pressure, density and dynamic viscosity, respectively.

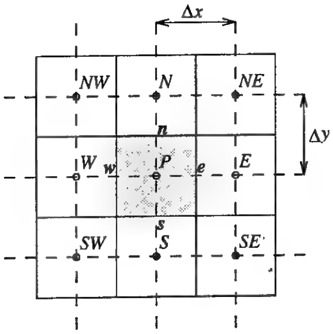


Figure 1: Control volume for nodal point P (upper and lower case denote nodal and face values, respectively).

When the first-order derivatives of equations (1) and (2) are integrated in the control volume P (Fig. 1) of a non-staggered grid, new equations will arise depending on the velocities at the faces of the control volume.

Face velocities

The relationship between the nodal and face values is found by discretization (in a control volume centred at the face of control volume P) of a simplified version of equation (2), obtained assuming mass conservation and constant viscosity

$$\rho\frac{\partial\left(U_iU_j\right)}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \mu\frac{\partial^2 U_i}{\partial x_j^2} . \tag{3}$$

Two different discretizations of the convective term of equation (3) were tested: (D1), with a first-order upwind scheme for the two derivatives; and (D2), with a second-order central difference scheme for the derivative in the direction perpendicular to the unknown velocity only. For instance, for velocity u_e we have in case of D1 discretization, if $u_e^{k-1} > 0$ and $v_e^{k-1} > 0$,

$$\begin{aligned}\rho\frac{\partial\left(U_iU_j\right)}{\partial x_j} &= 2\rho u_e^{k-1}\frac{u_e^k - U_P^k}{\Delta x}\\ &+ \rho v_e^{k-1}\frac{2u_e^k - U_S^k - U_{SE}^k}{2\Delta y} ,\end{aligned} \tag{4}$$

and in case of D2,

$$\begin{aligned}\rho\frac{\partial\left(U_iU_j\right)}{\partial x_j} &= 2\rho u_e^{k-1}\frac{u_e^k - U_P^k}{\Delta x}\\ &+ \frac{\rho v_e^{k-1}}{4}\left(\frac{U_N^k - U_S^k}{\Delta y} + \frac{U_{NE}^k - U_{SE}^k}{\Delta y}\right)\end{aligned} \tag{5}$$

if $u_e^{k-1} > 0$. The superscripts $k-1$ and k denote previous and current iteration.

The second member of equation (3) is discretized using second-order finite differences,

$$-\frac{\partial P}{\partial x} = -\frac{P_E^k - P_P^k}{\Delta x} , \tag{6}$$

$$\mu\frac{\partial^2 U}{\partial x^2} = 4\mu\frac{U_P^k - 2u_e^k + U_E^k}{\Delta x^2} , \tag{7}$$

$$\begin{aligned}\mu\frac{\partial^2 U}{\partial y^2} = \frac{\mu}{2}\Bigg(&\frac{U_N^k - 2U_P^k + U_S^k}{\Delta y^2}\\ &+ \frac{U_{NE}^k - 2U_E^k + U_{SE}^k}{\Delta y^2}\Bigg) .\end{aligned} \tag{8}$$

Replacing (4) – (8) into equation (3),

$$u_e = \sum_{nb} A_{nb}^U U_{nb}^k + \sum_{nb} A_{nb}^P P_{nb}^k , \tag{9}$$

where A_{nb}^U, A_{nb}^P are the coefficients for the nodal values of velocity and pressure surrounding face east.

Mass conservation

When equation (1) is integrated in the control volume of Fig. 1,

$$\int_{\mathcal{V}} \frac{\partial U_i}{\partial x_i}\partial\mathcal{V} = \left(u_e^k - u_w^k\right)\Delta y + \left(v_n^k - v_s^k\right)\Delta x \tag{10}$$

The equations for face velocities (9) are now replaced into (10) leading to the following algebraic equation,

$$\sum_{nb} A_{nb}^{UC} U_{nb}^k + \sum_{nb} A_{nb}^{VC} V_{nb}^k + \sum_{nb} A_{nb}^{PC} P_{nb}^k = 0 \tag{11}$$

The coefficients A_{nb}^{UC} and A_{nb}^{VC} represent links to a total of 18 nodal velocities, i.e. 9-node star for velocity U and V, surrounding P; whereas A_{nb}^{PC} includes connections to 5 nodal pressures (P_P, P_E, P_W, P_N and P_S).

Momentum conservation

The integration of equation (2) along $i = 1$ direction is,

$$\begin{aligned}&\rho\left(u_e^{k-1}u_e^k - u_w^{k-1}u_w^k\right)\Delta y +\\ &\quad\rho\left(v_n^{k-1}u_n^k - v_s^{k-1}u_s^k\right)\Delta x = -\frac{P_E^k - P_W^k}{2}\Delta y\\ &\quad + \mu\left(\frac{U_E^k - U_P^k}{\Delta x} - \frac{U_P^k - U_W^k}{\Delta x}\right)2\Delta y\\ &\quad + \mu\left(\frac{U_N^k - U_P^k}{\Delta y} - \frac{U_P^k - U_S^k}{\Delta y}\right)\Delta x\\ &\quad + \mu\left(\frac{V_{NE}^k - V_{NW}^k - V_{SE}^k + V_{SW}^k}{4\Delta x}\right)\Delta y ,\end{aligned} \tag{12}$$

which, after replacing the equations for the velocities at the faces, yields,

$$\sum_{nb} A_{nb}^{UU} U_{nb}^{k} + \sum_{nb} A_{nb}^{VU} V_{nb}^{k} + \sum_{nb} A_{nb}^{PU} P_{nb}^{k} = 0 \qquad (13)$$

The coefficients A_{nb}^{UU} represent links to the 9 nodal U velocities surrounding P, and A_{nb}^{VU} represent links to 4 nodal velocities (V_{NE}, V_{NW}, V_{SE} and V_{SW}). The A_{nb}^{PU} coefficients includes the contributions from 7 nodal pressures (P_P, P_E, P_W, P_{NE}, P_{NW}, P_{SE} and P_{SW}).

The momentum equation in the $i = 2$ direction

$$\sum_{nb} A_{nb}^{VV} V_{nb}^{k} + \sum_{nb} A_{nb}^{UV} U_{nb}^{k} + \sum_{nb} A_{nb}^{PV} P_{nb}^{k} = 0 \qquad (14)$$

may be obtained by an identical procedure.

Equations (11), (13) and (14) are all assembled in a single system of equations and solved simultaneously. The system of equations is of the form,

$$\mathbf{Ax} = \mathbf{b} \qquad (15)$$

$\mathbf{x}$ is a vector with sequence of blocks with variables U, V and P. The order of matrix A is $(NI-2) \times (NJ-2) \times K$, where $NI \times NJ$ is the problem size, and K stands for the number of variables (i.e., 3 in case of a two-dimensional laminar flow). This is a sparse, unsymmetric, block-band (block tridiagonal) linear system. Because this is the most time consuming part of the algorithm, special attention was given to this subproblem (in Section 3.3.1).

After solution of the linear system (15) one global iteration is completed. Because of the non-linearity of the differential governing equations, several global iterations are needed to obtain convergence and new coefficients are calculated using the new velocity and pressure fields, repeating the process until convergence. The nomenclature "global iteration" is used here to distinguish from the number of iterations associated with the solver.

Because all the conservation equations are solved as part of a single set, with no decoupling (or segregation, accordingly to nomenclature in ref. [11]), the algorithm can converge in a small number of iterations, and for this reason it has been designated DIRECTO [10] (direct, in English).

3 DISCUSSION OF RESULTS

The code development was made using the two classical geometries of a two-dimensional cavity with a sliding lid and a sudden expansion. In this paper results will be presented for the two-dimensional square cavity only.

This Section discusses 3 major aspects of the algorithm: the accuracy, memory requirements and computing time, in subsections 3.1, 3.2 and 3.3, respectively.

3.1 Accuracy

To obtain the accuracy of DIRECTO we performed simulations of the flow in a two-dimensional square cavity with sliding lid for 2 Reynolds numbers, (400 and 1000), and 3 grid sizes (64×64, 96×96 and 128×128). The Reynolds number definition was $Re = \rho U_{lid} H / \mu$. U_{lid} is the lid velocity and H is the size of the square cavity.

The velocities were set constant at every boundary, and zero normal gradient for the pressure was used. This condition was implemented in an implicit fashion to preserve the implicit feature of the method. The calculations were stopped for residuals lower than 1×10^{-5}. The residuals are the sum of the absolute errors of the algebraic equations divided by reference quantities $\rho U_{lid}^2 H$, and $\rho U_{lid} H$ for momentum and continuity equations, respectively. Calculations were all performed in single precision.

Method	Grid	U_{min}	V_{min}	V_{max}
DIRECTO D1	64	−0.31999	−0.43943	0.29404
	96	−0.32443	−0.44721	0.29897
	128	−0.32614	−0.44996	0.30090
Exact value		−0.32878	−0.45356	0.30399
Accuracy		1.73	1.97	1.69
DIRECTO D2	64	−0.31956	−0.43968	0.29383
	96	−0.32430	−0.44729	0.29893
	128	−0.32608	−0.45000	0.30087
Exact value		−0.32877	−0.45360	0.30380
Accuracy		1.78	1.95	1.77
CPI	64	−0.32368	−0.44862	0.29925
	96	−0.32653	−0.45163	0.30183
	128	−0.32751	−0.45274	0.30271
Exact value		−0.32873	−0.45431	0.30379
Accuracy		2.05	1.85	2.07
SIMPLE	128	−0.32614	−0.45119	0.30143

Table 1: Square cavity results for Re = 400 (CPI results from Deng *et al.*, 1994).

The estimated exact values and order of accuracy of the results were estimated following the generalization of the Richardson extrapolation method. The exact value can be approximated in terms of results on finite grids plus the leading term of the truncation error as,

$$\phi_1 = \phi_{ex} + h_1^n X_n + \dots, \qquad (16)$$

$$\phi_2 = \phi_{ex} + h_2^n X_n + \dots, \qquad (17)$$

$$\phi_3 = \phi_{ex} + h_3^n X_n + \dots, \qquad (18)$$

where h is the grid spacing in both directions and X_n is a grid function, assumed the same for every grid spacing (h_1, h_2 and h_3). Provided that h is small enough for the leading term to be dominant, the order of the numerical scheme is estimated as [17],

$$n = \frac{\ln\left[\mathrm{A}\,(h_2/h_1)^n - \mathrm{A} + 1\right]}{\ln\,(h_3/h_1)}, \qquad (19)$$

with,

$$A = \frac{\phi_3 - \phi_1}{\phi_2 - \phi_1} \qquad (20)$$

Tables 1 and 2 show non-dimensioned values of maximum and minimum V velocity (V_{max} and V_{min}) in the horizontal centreline and minimum value of U in the vertical centreline. The tables include values predicted by 3 different grids and also the order of accuracy and estimated exact value, obtained by the Richardson extrapolation. Results are also shown for SIMPLE algorithm of Patankar [1, 2], CPI (*Consistent Physical Interpolation*) and CSG (*Centred Staggered Grid*) methods of Deng *et al.* [17]. D1 and D2 are versions of DIRECTO algorithm with the convective term of equation (3) discretized using equations (4) and (5), respectively.

Method	Grid	U_{min}	V_{min}	V_{max}
DIRECTO D1	64	−0.36722	−0.49426	0.35565
	96	−0.37763	−0.51104	0.36628
	128	−0.38198	−0.51747	0.37037
Exact value		−0.38510	−0.52146	0.37292
Accuracy		1.26	1.38	1.38
DIRECTO D2	64	−0.36544	−0.49382	0.35414
	96	−0.37704	−0.51088	0.36577
	128	−0.38177	−0.51747	0.37022
Exact value		−0.39003	−0.52772	0.37701
Accuracy		1.57	1.73	1.75
CPI	64	−0.37436	−0.51015	0.36364
	96	−0.38233	−0.51947	0.37109
	128	−0.38511	−0.52280	0.37369
Exact value		−0.38867	−0.52724	0.37702
Accuracy		2.01	1.94	2.01
CSG	64	−0.35726	−0.48858	0.34556
	96	−0.37441	−0.50982	0.36271
	128	−0.38050	−0.51727	0.36884
Exact value		−0.38855	−0.52690	0.37705
Accuracy		1.96	1.94	1.99
SIMPLE	128	−0.37233	−0.51014	0.36234

Table 2: Square cavity results for Re = 1000 (CPI and CSG results from Deng *et al.*, 1994).

The SIMPLE algorithm [1, 2] is implemented here in a non-staggered grid [18] [19] and uses the hybrid finite difference scheme, switching from upwind to central differencing for mesh Reynolds number higher than 2.

The CPI method of Deng *et al.* [17] is similar to the DIRECTO method. CPI uses a governing differential equation for momentum and a relationship between face and nodal values identical to our equations (3) and (4), designated D1. In CPI and CSG, the governing differential equations of momentum are discretized at the centre of the control volumes before integration, while in the DIRECTO method the equations are first integrated in the control volume and then discretized. This leads to different velocity coefficients in discretized momentum equation of CPI and DIRECTO for control volumes near the boundaries. Differences could also occur in the implementation of the pressure boundary conditions; Deng *et al.* [17] does not state explicitly what kind of boundaries were used.

In the CSG method of Deng *et al.* [17] the convection terms are discretized using central finite differences for all mesh Reynolds numbers ($\rho U_i \Delta x_i / \mu$) and a staggered grid is used, to avoid velocity interpolation when discretizing the mass conservation equation.

As can be seen in Table 1, the estimated exact values from D1, D2 and CPI methods agree well with each other. The maximum percentual error is 0.17%, and is found in the V_{min} for the D1 method. The estimated accuracy of D1 and D2 is almost second order, although lower than CPI method (if based on U_{min} or V_{max}).

In case of Re=1000, Table 2, the decrease of accuracy of the D1 method is obvious. This is due to the first-order upwind scheme used to discretize the convective term of equation (3), as can be seen by comparison with the D2 method, which uses a second-order accurate finite difference discretization. Nevertheless the estimated accuracy of CPI is closer to second order compared with D2. The largest difference between the estimated exact values produced by D2 and CPI is 0.35% (based on U_{min}).

Compared to DIRECTO, SIMPLE algorithm with the hybrid scheme requires finer grids to achieve identical level of accuracy, as can be seen in Fig. 2 and Table 2. At Re=1000, the results of D1 and D2 methods for a grid of 96×96 are much closer to the estimated exact value of the CPI method than the results of SIMPLE with a grid of 128×128. This lack of accuracy is due to hybrid scheme that is first-order accurate for Peclet numbers greater than 2.

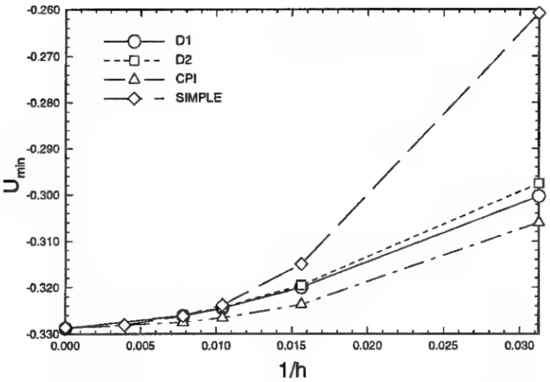


Figure 2: Evolution of U_{min} with grid resolution for Re=400.

Fig. 3 shows the streamlines for D2 and SIMPLE methods for a Re=1000 and a grid of 96×96. SIMPLE is unable to predict the streamline distribution in the centre. The dashed line (SIMPLE) at the centre of the flow represents the the value −0.11, whereas the solid line (D2) represents −0.115.

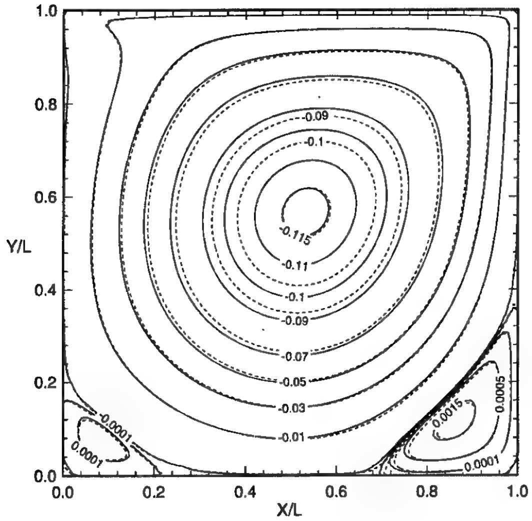


Figure 3: Stream function for D1 method (—) and SIMPLE method (– –) at Re=1000.

Given the similarities between DIRECTO and the algorithms CPI and CSG of Deng *et al.* [17] one would expect higher accuracy of the DIRECTO algorithm; this is an aspect requiring further attention.

3.2 Memory Requirements

Fig 4 shows the memory requirements for different implementations of DIRECTO, compared to SIMPLE using hybrid differencing.

The coefficient matrix, derived from a 9-node star, has a dimension of $(NI-2)\times(NJ-2)\times K$ by $(NI-2)\times(NJ-2)\times K$, where $NI\times NJ$ is the problem size, and K stands for the number of variables (i.e., 3 in case of a two-dimensional laminar flow). This is shown by line a) in Fig. 4.

Because of the block-tridiagonal structure it can be stored as a $[(NI-2)\times(NJ-2)]\times K$ by $2\times[(NI-2)\times 3+5]+1$ matrix (b) in Fig. 4).

For finer grids the block band structure becomes sparser. This was exploited by using a sparse matrix structure, storing only the non-zeros values on a vector, the column indices on an integer vector and using pointers to the beginning of each row. This structure reduced the memory requirements to $[(NI-2)\times(NJ-2)\times K]\times K\times 9$ (line c) in Fig. 4).

On the other hand SIMPLE only requires 5 matrices of dimension $(NI-2)$ by $(NJ-2)$ to store the coefficients (d) in Fig. 4).

3.3 Computing Time

The following computer tests were run on a DEC AlphaStation AXP 3000, model 600S, for sequential algorithms and a cluster of 4 DEC AlphaStations AXP 3000, models 500S and 600S, connected by FDDI and Gigaswitch using PVM, for parallel versions.

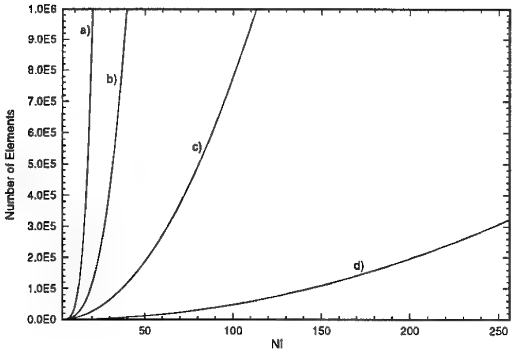


Figure 4: Memory requirements of DIRECTO (a), b) and c)) compared to SIMPLE (d)) using hybrid finite difference discretization scheme.

3.3.1 Linear solvers

The Gaussian elimination method was used initially [10] during the FORTRAN implementation of the algorithm. The first idea was to optimize the Gaussian elimination method by adapting it to the block band structure and using BLAS kernels and LAPACK library [20], on a vector processor VAX 6520-2VP. This reduced the computing time but still far from the SIMPLE+TDMA method, and required a large amount of storage [13].

The next stage was the use of an iterative method so that the sparse structure could be taken into account. An iterative method has the additional advantage of controlling the degree of accuracy for solving the linear system of equations. Because the solver is an inner step of a global iteration required because of nonlinearities, solving the equations to a high degree of accuracy may prove useless.

Several methods were tested and GMRES (Generalized Minimum Residual) [14] was retained for its robustness. GMRES is a Galerkin type method based on an orthonormal basis of a Krylov subspace. To obtain the solution of

$$\mathbf{Ax} = \mathbf{b}\,,$$

of the form

$$\mathbf{x}_k = \mathbf{x}_0 + \mathbf{z}_k\,, \tag{21}$$

where $\mathbf{x}_0$ is an initial solution with residual

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0\,. \tag{22}$$

$\mathbf{z}_k$ is computed such that its residual projected onto the Krylov subspace generated by $\mathbf{r}_0$ is minimized. Iterative methods of this type require the use of preconditioners in order to improve the convergence rate. Several preconditioners were tested and the best proved to be the Incomplete LU factorization of degree zero ILU(0) and ILUT [14, 21]. The diagonal preconditioner although very simple to implement did not give as good results as the others [12].

Table 3 shows the total CPU times, and corresponding number of outer iterations needed to achieve a residual of 1×10^{-5}, on a DEC AlphaStation AXP 3000 model 600S, for several grid sizes, using the DIRECTO+GMRES methods and for the SIMPLE+TDMA method [22]. The SIMPLE algorithm was used with 4 sweeps of TDMA for computation of the velocities and 8 for computation of the pressure. It can be seen in Table 3 that the CPU times are competitive. ILU(0) is a good choice for small grid sizes and ILUT is recommended for finer grids because it keeps the number of outer iterations low.

Grid	SIMPLE TDMA	D/GMRES ILU(0)	D/GMRES ILUT
32	13.1s (124)	8.5s (19)	11.7s (16)
64	169.9s (381)	99.4s (20)	89.2s (16)
96	764.2s (735)	608.4s (39)	285.2s (23)
128	2292.0s (1212)	10689.1s (91)	1667.2s (65)

Table 3: CPU time and number of iterations for a residual of 1×10^{-5}.

3.3.2 Parallelization

For this type of problems the parallelization by domain decomposition was selected. A non-overlapping domain decomposition strategy was used, where the domain was decomposed into disjoint subdomains separated by interfaces. The grid nodes were numbered first inside each subdomain and then on the interfaces, leading to a bordered block diagonal matrix shown in Figs. 5 and 6 [23].

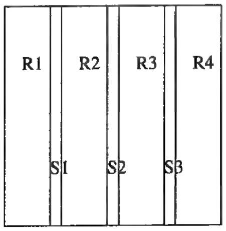


Figure 5: One-way dissection ordering.

The algorithm was parallelized in two versions: the first using a master-slave approach where the master performs the computations corresponding to the interface, and the second using a SPMD (Single Program Multiple Data) approach where each processor deals with a subdomain and one interface. Each processor had an independent preconditioner.

Table 4 reports the CPU and elapsed times on a cluster of workstations for a 64×64 grid and 3, 4 and 5 processes. There was a reduction in (elapsed) time, when passing from 3 to 4 processes; this is a 22% reduction corresponding to a relative speed-up of 1.27. For 5 processes, there is a degradation of CPU and elapsed times because the farm is composed only by 4 machines, and more than 1 process will have to share the same processing element.

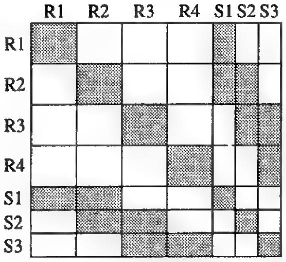


Figure 6: Reordered (one-way dissection) matrix.

	CPU time		Elapsed time
Processes	Master	Slave (max.)	-
3	192.7s	2734.3s	3090.4s
4	627.7s	1780.3s	2434.5s
5	1167.7s	1217.4s	3401.5s

Table 4: CPU and elapsed time for a 64×64 grid and Master-Slave approach.

To be able to use finer coarse-grain parallelism it is necessary to reduce the CPU time spent by the master to accompany the decreasing of the total time induced by the reduction of the CPU time in the slaves. Based on this need, another parallel version of the code was created, based on a SPMD strategy. Table 5 shows the CPU and elapsed times of the Master-Slave and SPMD approaches for 3 subdomains on a 64×64 grid. The SPMD approach is faster because it is more adequate to the sparse nature of the problem. Furthermore, for identical number of subdomains, the SPMD approach uses one process less than the Master-Slave. However the implementation of the SPMD approach is more complex, and given the reduced number of workstations available running PVM, we cannot conclude yet which of these approaches is the most appropriate.

	CPU time	Elapsed time
Master-Slave	1780.3s	2434.5s
SPMD	1568.5s	1642.3s

Table 5: CPU and elapsed times for a 64×64 grid for Master-Slave and SPMD approaches.

4 CONCLUSIONS

The present article reported on further developments of an implicit coupled algorithm for fluid flow equations. The main conclusions of this study are the following.

1. Computing time reduction was achieved by moving from a direct to an iterative solver based on GMRES.

2. It was shown that DIRECTO + GMRES with ILU(0) and ILUT preconditioners is always faster than SIMPLE+TDMA (4 and 8 sweeps for velocities and pressure, respectively). ILU(0) is a good preconditioner for coarse grids and ILUT is better for finer grids, because it keeps the number of outer iterations small.

3. Reduction of memory storage was also achieved by taking advantage of the sparse nature of the coefficient matrix. However, memory requirements are still large compared to SIMPLE and this is an aspect calling for further investigation. Efforts were made to overcome this disadvantage by reverting to domain decomposition techniques, at the expense of increased computing time.

Acknowledgments

This work was developed as part of project STRIDE N. STRDA/C/CEG/712/92. F.A. Castro is a PhD student, recipient grants from Programmes *CIÊNCIA* and *PRAXIS XXI*. The work on preconditioners was carried out during the visits of two of us to CERFACS on July 1994. F.D.A. and P.V. thank the hospitality of Prof. Ian Duff.

The authors are grateful to the Faculty of Engineering Computer Centre (CICA).

References

[1] S. V. Patankar and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transfer*, 15:1787–1806, 1972.

[2] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow.* Hemisphere Publishing Corporation, 1980.

[3] G.D. Raithby and G.E. Schneider. Numerical solution of problems in incompressible fluid flow: treatment of the velocity pressure coupling. *Numerical Heat Transfer*, 2:417–440, 1979.

[4] A. Wanik and U. Schnell. Some remarks on the PISO and SIMPLE algoritms for steady turbulent flow problems. *Computers and Fluids*, 17(4):555–570, 1989.

[5] D.S. Jang, R. Jetli, and S. Acharya. Comparison of the PISO, SIMPLER and SIMPLEC algorithms for the treatment of the pressure-velocity coupling in steady flow problems. *Numerical Heat Transfer*, 10:209–228, 1986.

[6] J.J. McGuirk and J.M.L.M. Palma. The efficiency of alternative pressure-correction formulations for incompressible turbulent flow problems. *Computers and Fluids*, 22(1):77–87, 1993.

[7] R.I. Issa. Solution of the implicity discretized fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62:40–65, 1986.

[8] S.V. Patankar. A calculation procedure for two-dimensional elliptic situations. *Numerical Heat Transfer*, 4:409–425, 1981.

[9] J.P. van Doormaal and G.D. Raithby. Enhancements of the SIMPLE method for predicting incompressible fluid flows. *Numerical Heat Transfer*, 17:147–163, 1984.

[10] F.A. Castro. A coupled procedure for solving the Navier-Stokes equations (in portuguese). Technical report, Faculty of Engineering, University of Porto, Portugal, 1993.

[11] J.P. van Doormal and G.D. Raithby. An evaluation of the segregated approach for predicting incompressible fluid flows. 1985. Presented at the National Heat Transfer Conference, Denver, Colorado, USA, ASME paper 85-HT-9.

[12] P.B. Vasconcelos and F.D. d'Almeida. Direct and iterative methods in coupled discretization of fluid flow problems. *Paper presented at LANCZOS Centenary Conference, Raleigh, North Carolina, USA, 12-17 December*, 1993.

[13] P.B. Vasconcelos and F.D. d'Almeida. Columnwise block LU factorization using BLAS kernels. *Computer Systems in Engineering*, 1995 (to appear).

[14] Y. Saad. GMRES: A Generalized Minimum Residual algorithm for solving nonsymetric linear systems. *SIAM, J. Sci. Statis. Comput.*, 7:856–869, 1986.

[15] A. Beguelin, J.J. Dongarra, G.A. Geist, R. Manchek, and V.S. Sunderam. A user's guide to PVM parallel virtual machine. Technical report, ORNL/TM-11826, Oak Ridge National Laboratory, July 1991.

[16] P. B. Vasconcelos. Experience with PVM and ScaLAPACK (in portuguese). Technical report, CICA-I01/94, Faculdade de Engenharia do Porto, 1994.

[17] G. B. Deng, J. Piquet, P. Queutey, and M. Visonneau. A new fully coupled solution of the Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 19:605–639, 1994.

[18] C. M. Rhie and W. L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21:1525–1532, 1983.

[19] T. F. Miller and F. W. Schmidt. Use of a pressure-weighted interpolation method for the solution of the incompressible Navier-Stokes equations on a nonstaggered grid system. *Numerical Heat Transfer*, 14:212–233, 1988.

[20] E. Andersom *et. al.* *LAPACK Users Guide.* SIAM, Philadelphia, 1992.

[21] R. Barrett *et. al.* *Templates.* SIAM, Philadelphia, 1994.

[22] P.B. Vasconcelos and F.D. d'Almeida. The effects of preconditioning iterative methods in coupled discretization of fluid flow problems. *16th Biennal Conference on Numerical Analysis*, University of Dundee, 27-30 June, 1995.

[23] P.B. Vasconcelos and F.D. d'Almeida. Parallel computation on fluid dynamics using PVM and domain decompositon techniques (in portuguese). *Proceedings of IV Nacional Mecânica Computacional.*, Vol.2, 815-824, 1995.

Experiments with Unstructured Grid Computations

S.V. Ramakrishnan, K.Y. Szema, C.L. Chen, V.V. Shankar
and S.R. Chakravarthy
Rockwell Science Center
1049 Camino Dos Rios
Thousand Oaks, CA 91360
U.S.A.

1. SUMMARY

This paper describes work done at Rockwell Science Center on the development and application of computational fluid dynamics (CFD) solvers for unstructured grids. A description of the use of "interior boundary" conditions in simulating moving bodies is also presented.

2. INTRODUCTION

The CFD group at Rockwell Science Center has been involved over the past fifteen years in the development and application of numerical techniques for the simulation of flow past complex aerodynamic shapes. Starting with small perturbation equations, codes have been developed to solve more and more complex governing equations on structured grids (Ref. 1-4). The latest version of the structured grid code solves Reynolds Averaged Navier-Stokes (RANS) equations in generalized curvilinear coordinates. It includes the ability to simulate reacting multispecies flows (Ref. 5). Simulations requiring grid movements are handled quite elegantly using this code (Ref. 6). CFD codes developed at Rockwell Science Center have played a significant role in several national projects including the Space Shuttle, B-1B and National AeroSpace Plane (NASP) projects.

Time required for performing accurate numerical simulation of complex fluid-dynamics problems is still sufficiently large to discourage designers from including CFD techniques in the design cycle. Total time required for a numerical simulation consists of the time required for

a) preprocessing, which consists of modifying the CAD geometry to a form suitable for numerical simulation, (in the case of structured grids) dividing the computational domain into zones, choosing proper grid resolution at the boundaries and finally grid generation,

b) solver,

and

c) post-processing, which consists of extracting physical quantities like skin-friction and heat-transfer, from the numerical solution; and visualization of the solution.

Several years of research in structured-grid simulations and developments in computer software and hardware technologies have considerably reduced the turnaround time for numerical solutions. Still the time required to simulate flow past complex geometries is unacceptably large. Especially, the time required for preprocessing increases almost exponentially as more and more details of the geometry are included in the simulation. For example, in the case of the multibody space shuttle configuration (Ref. 7), several months were needed to generate a structured grid when the fidelity requirements for the model employed in the numerical simulation were increased considerably.

Unstructured grid methodologies appear to be very promising, since the preprocessing time could be orders of magnitude less than that required for structured grids. It is indeed the case for inviscid flows. But, our experience with unstructured grid computations has opened our eyes to several issues involved in such simulations. We propose to discuss some of those issues in this paper.

Research on the development of unstructured grid solvers for Computational Fluid Dynamics (CFD) and Computational ElectroMagnetics (CEM) has been in progress at the Rockwell Science Center for the past several years. An unstructured grid solver for CFD, called UNIV, that can handle tetrahedral, triangular prizmatic and hexahedral cells, has been developed (Ref. 8). UNIV employs a finite-element-like formulation that uses piecewise polynomial interpolation for the dependent variables. The dependent variables are the cell averages of internal energy, mass, x-, y-, and z-momenta. Interpolating polynomials may be discontinuous across cell boundaries. An approximate Riemann solver is used to resolve discontinuities at cell boundaries. The domain of a dependent variable polynomial is restricted to a cell. The discretization of the governing equations is constructed directly from the integral form of the conservation laws. No variational principle or method of weighted residuals or other indirect approach is employed. The code has the option to use either a least-square polynomial or a ENO (Essentially Non-Oscillatory) reconstruction. Reconstruction is the process of constructing an interpolating function for a cell that satisfies the cell average. Please see Ref. 9. for details on ENO schemes.

Numerical formulation employed in UNIV and a new approach for simulating bodies in relative motion are discussed in the following sections. A generalized Lax-Wendroff scheme for Euler equations adapted from CEM is also presented. A pointwise turbulence model that is highly suitable for unstructured grids is discussed. Lessons learned from our experience with unstructured grid computations are elucidated.

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

3. NUMERICAL FORMULATION

Two different approaches to solving the initial-/boundary-value problem (IBVP) for general hyperbolic system of conservation laws in the "conservation-law form" represented by

$$\frac{\partial q}{\partial t}+\frac{\partial f_1}{\partial x}+\frac{\partial f_2}{\partial y}+\frac{\partial f_3}{\partial z}=0 \qquad (1)$$

have been developed. Equation (1) is satisfied at all (*x, y, z*) belonging to domain *D* with prescribed initial and boundary values for the dependent (conserved) variable vector *q*. Here, the Cartesian coordinate directions (independent variables) are *x, y*, and *z*. The components of flux tensor in the three coordinate directions are the vectors f_1, f_2 and f_3. In both approaches the domain *D* is divided into several cells, and the integral form of the conservation equations in each cell given by

$$\frac{\partial}{\partial t}(\bar{q}V)+\int\int_S\left(\vec{F}\cdot\hat{n}\right)dS=0 \qquad (2)$$

is solved with prescribed initial values for $\bar{q}$

$$\bar{q}_0=\bar{q}(x,y,z,t_0)$$

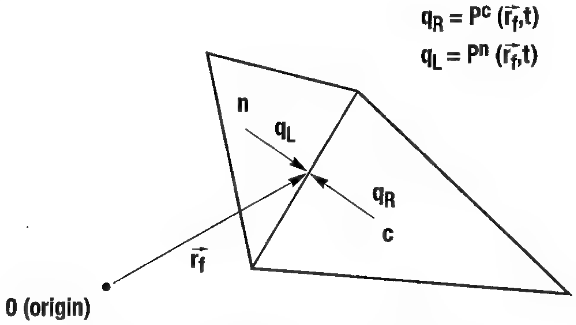
and relevant boundary conditions. Here, $\bar{q}$ denotes the cell average of the dependent variables;

$$\hat{n}=\hat{n}_x\vec{j}+\hat{n}_y\vec{k}+\hat{n}_z\vec{l}$$

is the outward unit normal at any point on the boundary surface *S* of a cell; $\vec{j}$, $\vec{k}$, and $\vec{l}$ are the unit vectors in *x*, $\vec{y}$ and *z* directions respectively; *V* is the cell volume and $\vec{F}$ is the tensor of fluxes with (f_1, f_2, f_3) as components. Stated in words, Eqn. 2 implies that the rate of increase of a conserved quantity ($\bar{q}V$) inside a cell is given by the net inflow (flux) of that quantity into the cell. Therefore, as in the case of cell-centered finite-volume structured grid solvers (Ref. 3), solving the governing equations requires evaluation of surface integrals from known values of cell averages.

Surface integrals are evaluated using numerical quadrature formulas. In this method an integral is written as the weighted sum of the integrant evaluated at the quadrature points. The location and weights of quadrature points are so chosen as to give the best possible approximation for the integral. Higher order schemes require larger number of quadrature points. Choosing the centroid of a surface as the quadrature point yields second order accuracy. Since only the cell-averages of the dependent variables are known, we need to develop a procedure for evaluating the dependent variables at the quadrature points in order to compute the surface integrals (fluxes). The spatial accuracy of the numerical scheme is determined by the accuracy of this "reconstruction" procedure. The dependent variable vector *q* at a quadrature point may not be uniquely specified, since the point belongs to two neighboring cells with different polynomial representations. If the two vectors evaluated at a quadrature point using the polynomial reconstruction in the two "containing" cells are q_L and q_R (Fig. 1), then a unique value q^* is determined from the solution of a locally one-dimensional Riemann problem with q_L and q_R as the "left" and "right" values. An approximate Riemann solver suggested by Roe (Ref. 10) is employed for this purpose in the UNIVERSE-series of codes of which UNIV is a member.

The two approaches alluded to at the beginning of this section differ in their "reconstruction" procedure and also in the time-stepping scheme. Only explicit time-stepping schemes are considered in both approaches. Both approaches permit use of multiple quadrature points and curved surfaces for higher accuracy. Codes developed using these approaches can handle hexahedral, tetrahedral and triangular- prismatic cells.



SC.1654E.091195

Fig. 1 "Left" and "Right" states for locally one-dimensional Riemann problem.

3.1 The First Approach; a Finite-element Like Algorithm

The major credit for this work goes to Dr. Chakravarthy. This approach employs a unified treatment for structured and unstructured grids. The codes developed using this formulation are called UNIVERSE-series of codes. The UNIVERSE-series includes "least-square" and "ENO" (Ref. 9) reconstruction options. Both these procedures involve development of an interpolating polynomial $P^C(x,y,z)$ for each of the conserved quantities, where

$$P^C(x,y,z)=\sum_{i=0}^{np} p_i^C\, x^{j(i)}\; y^{k(i)}\; z^{l(i)} \qquad (3)$$

where, p_i^C are the coefficients of the polynomial. P^C is applicable only within a given cell *C*. Integral of P^C over *C* reproduces the corresponding cell average. That is,

$$\frac{1}{V}\iiint_C P^C dV=\sum_{i=0}^{np} a_i^C p_i^C=\bar{q}^C \qquad (4)$$

where,

$$a_i^C=\frac{\iiint_C x^{j(i)}\,y^{k(i)}\,z^{l(i)}dV}{\iiint_C dV} \qquad (5)$$

The spatial accuracy of the numerical scheme is determined by the form of P^C. A linear polynomial in *x, y* and *z* results in second-order accuracy while a quadratic polynomial yields a third- order scheme. Linear polynomial requires evaluation of 4 coefficients, while the quadratic polynomial requires 10. In the case of the "least-square" option, the polynomial coefficients are computed such that the integral of P^C over cell C reproduces the corresponding cell average values (Eqn. 4), and the integrals over the neighboring cells satisfy the corresponding cell averages in a least-square sense. That is,

$$\frac{\partial}{\partial p^C_i} E = 0 \qquad (6)$$

for $0 \le i \le np$. The error term E is given by,

$$E = \sum_{n=1}^{nc} \left(\sum_i a^n_i p^C_i - \bar{q}\, n \right)^2 \qquad (7)$$

where *n* refers to a neighboring cell, *nc* is the number of cells in the neighborhood of cell *C*, excluding *C* itself. Obviously, a least-square approximation for P^C can be constructed only if

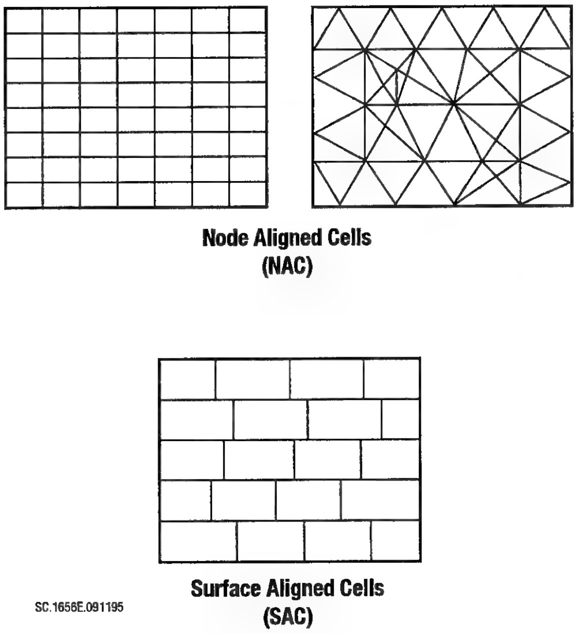
$$nc \ge np \qquad (8)$$

Therefore, the neighborhood of a cell should be properly defined to satisfy equation (8). The UNIVERSE-series CFD formulation defines a "neighbor" of a given cell in a very flexible and useful way.

First, we consider two types of cell connectivities (Fig. 2):

(1) Node-aligned cells (NAC)

(2) Surface-aligned cells (SAC)


Node Aligned Cells (NAC)

Surface Aligned Cells (SAC)

Fig. 2 Node-aligned and surface-aligned cells.

Next, we consider different types of neighbors:

(1) Touching neighbors (TN)

These include

(1a) Common-node neighbor (CNN)

(1b) Common-face neighbor (CFN)

(1c) Touching-face neighbor (TFN)

(2) Proximity neighbors (PN)

This latter type is defined in terms of distance from a given cell.

A neighborhood is now defined to be a collection of neighboring cells. A neighborhood hierarchy is defined as follows:

H^0 is the cell itself.

H^1 is the cell and its neighbors.

H^2 is the union of H^1 and the neighbors of all the cells in H^1.

This process may be continued recursively, and depending on the order of P^C, a neighborhood may be found such that equation (8) is satisfied.

In the case of ENO (Essentially NonOscillatory) reconstruction, we seek to obtain a "best" polynomial rather than a "least-squares" one. The "best" polynomial corresponds to the "smoothest". As always, the equation for cell *C* must be satisfied (Eqn. 4). From the remaining *nc* equations, we can select any combination of *np* equations and solve the resulting set of *np + 1* equations. There are

$$\binom{nc}{np} \qquad (9)$$

such combinations. The combination that yields the best polynomial in terms of its ENO property is to be preferred. For example, when the flow field contains a single shock wave, the neighbors selected should lie on the same side of the shock as cell *C*. This approach may be termed the "best stencil" formulation and has been applied very successfully in various forms to structured grid ENO formulations. Reference 9 contains many different strategies for this task. Note that the "least squares" strategy may result in a stencil that includes cells from both sides of a discontinuity and hence not desirable.

Alternatively, a "best term" strategy has also been tried out. In this formulation, the least-squares polynomials are first determined for all cells. Each coefficient p^C_i of the polynomial P^C (Eqn. 3) in a given cell *C* corresponds to the appropriate derivative of the polynomial (up to a constant coefficient) evaluated at the centroid of the cell. That is,

$$\left(\frac{\partial P^C}{\partial x^{j(i)} \partial y^{k(i)} \partial z^{l(i)}} \right)_c = K^i p^C_i \qquad (10)$$

where the subscript *c* refers to the centroid and K^i is a constant. In the case of the "best term" strategy, we replace each p^C_i by that computed from the corresponding derivative at the cell centroid evaluated from a neighboring cell polynomial p^n_i, provided

$$\left|p^n_i\right| < \alpha \left|p^C_i\right| \qquad (11)$$

where $\alpha > 1$. In other words, p^C_i is selected such that the corresponding derivative at the centroid does not differ "too much" from its value in a neighborhood. This procedure attempts to construct a reconstruction polynomial that uses only neighboring cells on the same side of a discontinuity.

For one-dimensional shock-tube problems, it has often been demonstrated that it is better to select the best stencils based on comparing interpolates of local characteristic variables and not the conserved dependent variables. However, within the context of unstructured grid formulations this approach is very expensive, and consideration of such issues is postponed for future work.

We have so far discussed the spatial discretization problem. As far as temporal discretization is concerned, only explicit schemes have been considered. A second-order time-accurate formulation is given below as an example. This is fashioned after Heun's method or the second-order Runge-Kutta method (RK2). The RK4 method can be implemented in similar fashion. Higher than second-order spatial accuracy results in reduced numerical dissipation, and this sometimes necessitates the use of the fourth-order Runge-Kutta formulation, which has a larger stability range than the second-order Runge-Kutta method.

In semidiscrete form, the equations to be solved are

$$\frac{\partial}{\partial}(\bar{q}\,V)=\mathrm{RHS}(\bar{q},t) \tag{12}$$

where $\mathrm{RHS}(\bar{q},t)$ is the net flux. The corresponding time stepping method can be written as

$$\begin{aligned}(\bar{q}\,V)^{1} &= (\bar{q}\,V)^{n}+\mathrm{RHS}(\bar{q}^{n},t^{n})\Delta t\\ (\bar{q}\,V)^{n+1} &= \frac{1}{2}\left[(\bar{q}\,V)^{n}+(\bar{q}\,V)^{1}+\Delta t\,\mathrm{RHS}(\bar{q}^{1},t^{n+1})\right]\end{aligned} \tag{13}$$

The fourth-order accurate Runge-Kutta scheme can be written as

$$\begin{aligned}k_1 &= \mathrm{RHS}(\bar{q}^{n},t^{n})\\ (\bar{q}\,V)^{1} &= (\bar{q}\,V)^{n}+\frac{\Delta t}{2}k_1\\ k_2 &= \mathrm{RHS}(\bar{q}^{1},t^{n}+\frac{\Delta t}{2})\\ (\bar{q}\,V)^{2} &= (\bar{q}\,V)^{n}+\frac{\Delta t}{2}k_2\\ k_3 &= \mathrm{RHS}(\bar{q}^{2},t^{n}+\frac{\Delta t}{2})\\ (\bar{q}\,V)^{3} &= (\bar{q}\,V)^{n}+\Delta t\,k_3\\ k_4 &= \mathrm{RHS}(\bar{q}^{3},t^{n+1})\\ (\bar{q}\,V)^{n+1} &= (\bar{q}\,V)^{n}+\frac{\Delta t}{6}(k_1+2k_2+2k_3+k_4)\end{aligned} \tag{14}$$

In the above, the explicit dependence of *RHS* on *t* is useful for time-dependent problems where the boundary conditions or other behavior explicitly depend on time.

3.2 The Second Approach; Generalized Lax-Wendroff Scheme

This numerical scheme was originally developed under the leadership of Dr. Shankar (Ref. 11) for solving Maxwell's equations and later was adapted for Euler equations. This approach employs a multilevel time stepping scheme. The second-order scheme uses a two time-level discretization. The first fractional time-step employs first-order spatial (q_L and q_R are set equal to the corresponding $\bar{q}^n$) and temporal discretizations to compute $\bar{q}^{n+1/2}$. Here, the superscript *n* refers to the time-level *n*. For the second time-step, q_L and q_R are computed from the corresponding centroidal values of $\bar{\mathbf{q}}^n$ and $\nabla\bar{q}^n$ as

$$q_L(\text{or } q_R)=\bar{q}^n+\left(\vec{\mathbf{r}}_f-\vec{\mathbf{r}}_C\right).\ \nabla\bar{q}^n \tag{15}$$

where $\vec{\mathbf{r}}_f$ and $\vec{\mathbf{r}}_C$ refer to the position vectors of the centroids of the surface and cell, respectively, and

$$\left(\nabla\bar{q}^n\right)\simeq\frac{1}{V}\int_S \hat{n}q^*dS \tag{16}$$

where q^* is obtained from the Roe's approximate Riemann solver with q_L and q_R set equal to $\bar{q}$n. The algorithm described above may be considered as a generalization of Lax-Wendroff upwind integration, since it reduces to the Lax-Wendroff scheme for uniform rectangular hexahedral cells. Note that only details of the second order scheme are presented, and that extension to higher order schemes is indeed straight-forward, albeit tedious.

3.3 Computation of Viscous Fluxes

Viscous fluxes at a quadrature point on a cell face are computed as the mean of the corresponding contributions from the two adjacent cells that share the face. That is, the average of the derivatives computed from the polynomial reconstruction in the two adjacent cells are employed in the calculation. When the quadrature point lies on the boundary of the computational domain, the polynomial reconstruction employed is centered about the quadrature point.

4. BOUNDARY CONDITIONS

The implementation of boundary conditions ensures consistency in flux computations. That is, just like in the case of any interior cell boundary, computation of fluxes for a cell boundary that lies on the boundary of the computational domain involves determination of "left" and "right" states and Roe's approximate Riemann solver. The state that corresponds to the "outside" of the domain should satisfy the appropriate boundary conditions. For instance, when computing fluxes for a cell on the left boundary of the domain where inviscid tangency condition is to be satisfied, the "left" state should be such that the corresponding velocity vector should be tangential to the surface. This manner of imposing boundary conditions ensures that only the information at a boundary that corresponds to waves propagating in to the computational domain is actually used in the computation of fluxes.

4.1 Interior Boundary Condition

The concept of boundary conditions has been generalized to include specification of boundary conditions anywhere in the computational domain (Ref. 12). The part of the boundary condition that does not correspond to the actual boundary of the computational grid is referred to as "interior boundary conditions." In this case the user specifies, among several attributes, the coordinate location of each boundary point as well as a vector normal associated with the point. The need for the normal arises from the fact that even though interior boundary points are specifiable as individual points, they arise from boundary surfaces that they are a part of. It is the surface normal along with its location that describes the local geometry. Note that the surface in question could very well be a surface of discontinuity (a shock wave), and it may not be possible to assign unique values for the dependent variables at the corresponding boundary point. To account for such a situation, for every interior boundary location identified by the user, two interior boundary points are created and added to the data base of the UNIV flow solver. One of the added points has the normal pointing one way and the second point the other way (Fig. 3).

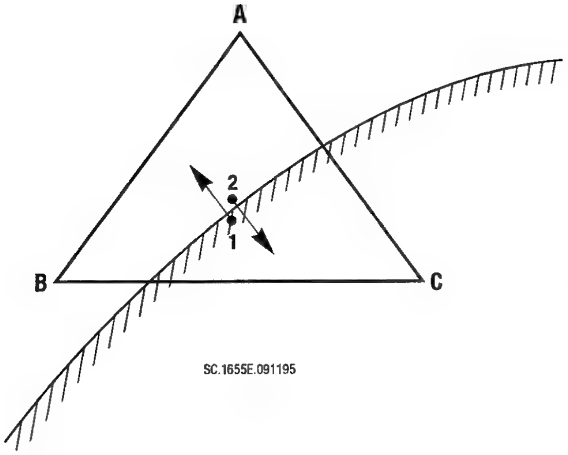


Fig. 3 Interior boundary points.

The user-specified boundary condition is applied to each pair of interior boundary points. Certain boundary conditions such as surface tangency are applied individually to both points of the pair; i.e., they are applied in a decoupled fashion. Certain boundary conditions such as those associated with "shock fitting" or "contact-surface fitting" are applied in a coupled fashion. For example, the values on the supersonic side of the shock are accepted as is, and the values on the subsonic side are computed (along with the shock speed value) by accepting only the pressure from the subsonic side and applying the Rankine-Hugoniot shock-jump relations. The availability of the boundary points in pairs facilitates such transactions.

The process of computing the "left" and "right" states is modified when a cell has an interior boundary point. The contribution of each of the boundary points to the quadrature points is computed using the proportion of the surface area that is in the region of influence of the boundary point. In Fig. 3, the face AB is completely in the region of influence of boundary point 1, whereas face CA gets contributions from both the boundary points. Note that the boundary points actually differ only in the normals associated with them, and their coordinates are identical. For the sake of clarity, they are shown as two different points in Fig. 3.

As part of the infrastructure necessary to implement interior boundary point treatment, one needs the ability to associate with each (pair of) interior boundary point the cell that contains it. This chore of searching through the mesh to determine the one cell that contains the boundary point is efficiently accomplished in the UNIV flow solver using an "octree" sort and search procedure. Given an interior boundary point, an octree search of the sorted list of node points of the mesh quickly yields the nearest mesh node. All cells that contain the node as well as the common-node neighbors of this set of cells are searched, in that order, to determine if the given point is in any of those cells. If not, the "nearest" cell is identified.

In the previous paragraphs, it was convenient to describe the procedure as if the user provides pointwise information related to interior boundaries. Depending on the relative fineness or coarseness of the geometry description of the interior surface with respect to the surrounding mesh, there may be two or more user-specified (before the flow solver replaces each user-specified point with two points, with the normals facing in opposite directions) interior boundary points in a cell, or there may be none (Fig. 4). In Fig. 4 the cells 1,5,8 and 10 have two or more interior boundary points while cells 4,7 and 9 have none. The case of multiple interior points in a cell can be dealt with easily (e.g., by replacing them with an equivalent single point, if necessary). But, the case of no interior point in a cell that actually straddles the interior boundary is not acceptable. To avoid such problems, we start with the user describing the interior surface as an unstructured grid (triangular elements). Using an octree-based sort and search procedure, the intersection of the mesh with this surface is identified (Fig. 5). Interior boundary points are assigned to each such intersection. There could be interior surface geometry elements that do not participate in such intersections. The centroids of these elements are optionally added to the list of interior boundary conditions.

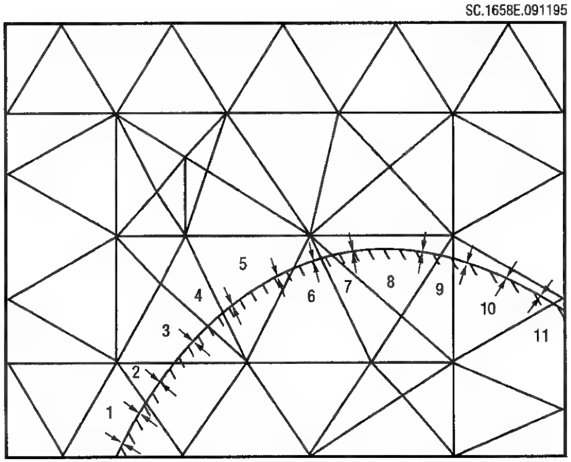


Fig. 4 An example of user specified interior boundary points.

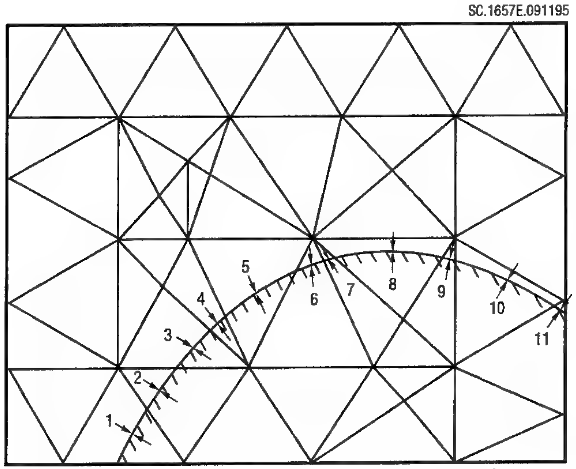


Fig. 5 An example of UNIV generated interior boundary points.

5. GRID GENERATION

The UNIVERSE-series of codes includes an unstructured grid generator, named UNIVG. UNIVG accepts specification of surface geometry in the form of a collection of patches. A patch geometry could be specified either in the IGES format or by specifying sufficient number of non-intersecting lines on the patch. Each line in turn is discretized by an ordered collection of sufficient number of points. Triangular elements are first generated on the boundary of the computational domain satisfying user specified clustering requirements. The computational domain is then discretized in the form of tetrahedral cells using the “advancing front” technique (Ref. 8).

The method of “advancing front” does not possess a good mechanism for controlling the distribution of cells in the computational domain, and often regions with large variations in cell sizes and shapes are encountered. Such regions deteriorate the fidelity of numerical simulation. To overcome this problem, grid smoothing strategies based on constrained optimization techniques have been developed and employed successfully (Ref. 13).

UNIVG has the capability to develop an unstructured grid that includes a specified “cloud” of points as nodes. This is sometime useful in controlling the distribution of cells in the computational domain.

6. STORE SEPARATION

The concept of interior boundary conditions, described in section 4.1, is used to simplify numerical simulation of the store separation problem. The process starts by generating an unstructured mesh for the parent vehicle. The store geometry is discretized by generating an unstructured surface mesh. The intersection of the store surface mesh with the unstructured volume cells of the parent grid is determined by using an octree sort/search procedure. Centroids of the intersecting surfaces and their normals are computed. A data base consisting of these centroids and normals is thus generated and used as input by the interior boundary condition routines. Note that the boundary condition for the store accounts for its initial motion. The governing equations are then solved with appropriate boundary conditions to obtain solution for the next time level. Aerodynamic forces and moments for the store are computed, and the equations for the conservation of linear and angular momenta are solved to obtain the location and orientation of the store at the next time level. This process also yields velocity vectors at all points on the store. The intersection of the store geometry in its new location with the unstructured volume mesh is determined, and the process is repeated for all subsequent time steps. This approach is not suitable for viscous flows, since the parent vehicle grid would be too coarse to resolve viscous regions when the store moves away from the parent vehicle. This problem may be circumvented by adapting the mesh as the store moves, but at present such a strategy does not appear attractive due to the large amount of work involved in adapting the mesh and performing required interpolations that could result in loss of accuracy.

7. TURBULENCE MODELING

Until recently all the turbulence models employed in numerical simulations required the knowledge of the normal distances of a point from surrounding walls. This information is very difficult to obtain in the case of unstructured grids. In the case of structured grids, mostly distances along grid lines were employed. This was sufficient since the grid lines were nearly orthogonal in the vicinity of a body where viscous effects are dominant. But when complex geometries requiring a multizone grid topology were encountered, it became difficult to maintain continuity of eddy viscosity at zonal interfaces. To circumvent this problem, a pointwise turbulence model that does not require any information regarding the distance of a point from surrounding walls was developed at Rockwell Science Center by Goldberg and Ramakrishnan (Ref. 14). Since then, several such models have been developed, and reliable computation of turbulent flows on structured grids has become a possibility.

8. LESSONS LEARNED

One of the most important lessons that we have learned from our own experience and the experience of our peers in the CFD and CEM community with unstructured grid computations is that in spite of all the advances that have been made in this field so far, on comparable grids structured-grid simulations yield more accurate solutions. For complex geometries, it is indeed possible to speed up the preprocessing stage of a numerical simulation by an order of magnitude by employing unstructured grids. On the other hand, a structured grid computation requires less CPU time and memory and converges in fewer number of time steps. For example, in the case of an inviscid flow past a sphere with $M_\infty = 0.5$, it has not been possible to obtain even two orders of magnitude drop in the L-2 norm for the net-flux vector in reasonable number of time steps (less than 1000) for an unstructured grid, while a structured grid computation on a comparable grid converges by about four orders of magnitude in less than 600 time steps. Attempts have been made to develop implicit schemes for unstructured grids, but in our opinion, a structured grid still performs better as far as convergence and accuracy are concerned.

During the design phase of an aerospace configuration, several possible candidates are evaluated, and a small number of viable candidates is down selected from the original pool for further considerations. This process is usually carried out using a relatively low-level CFD analysis requiring less stringent accuracy and convergence criteria. Mostly only inviscid flows are considered. Structured grids are not suitable for this purpose, since the preprocessing takes an unacceptably long time. Unstructured Euler solvers offer the most viable solution. Since Euler equations, unlike Navier-Stokes equations, do not require very fine grids in the vicinity of solid bodies, unstructured grid development becomes much easier to handle, and several solutions for many different configurations can be carried out in a matter of a few weeks. This was indeed demonstrated in the case of some modifications that were carried out for B-1B bomber. Starting with the geometry of the aircraft in IGES format, an Euler solution was obtained for this complex configuration (Fig. 6) in about five working days. With the use of Massively Parallel Processing (MPP) computers, this process may be accelerated even more. From this point of view, unstructured grid solvers have a clear edge over their structured grid counterparts.

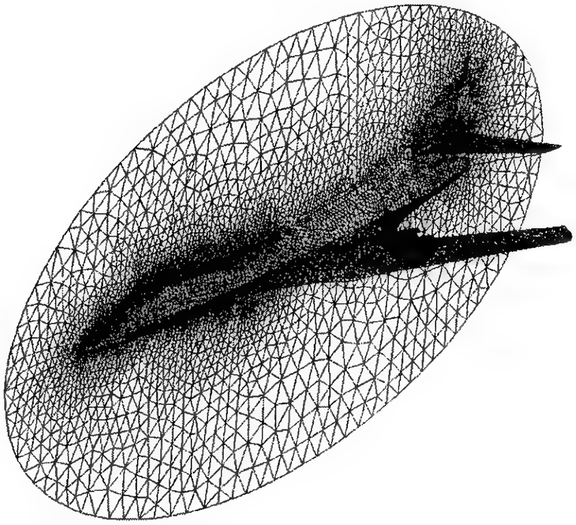


Fig. 6 Unstructured grid for inviscid flow past B-1B configuration.

In the case of viscous flows, stringent resolution requirements in the direction normal to a solid body force an unstructured tetrahedral grid to have similar resolutions on the body surface in order to maintain acceptable shapes for the conservation cells in such regions. This results in an unstructured grid with too many conservation cells in the vicinity of a solid body. Such a restriction does not exist in the case of a structured grid and thus makes it more suitable for viscous flows.

Arguably a hybrid grid, consisting of a structured grid in the viscous regions and unstructured grid elsewhere may be the most suitable way to discretize a computational domain. But, considering the fact that one of the main reasons for resorting to an unstructured grid is the difficulty involved in generating body-conforming structured grids for complex geometries, it is indeed questionable whether much could be gained from such a strategy. At the Rockwell Science Center we have been experimenting with a hybrid unstructured grid consisting of triangular prismatic cells in the viscous regions and tetrahedral cells elsewhere. This approach seems to be promising.

One aspect of unstructured-grid solvers in which real progress has been made is the storage requirement. Whereas the structured grid solvers require only about 30 words of storage per conservation cell, the unstructured grid solvers used to demand about 200. This situation has been vastly improved, and the storage requirement has been brought down to a manageable 60 words per conservation cell.

The concept of "interior" boundary conditions described in section 4.1 is very promising. It was used successfully in computing the trajectory of a store released from an F-18. This concept also proved its usefulness in analyzing the effect of mounting an additional equipment on an aircraft. In this case, the grid and solution from an earlier computation could be used along with the geometry of the added equipment to obtain the required information in a timely manner. The present implementation of this concept has some shortcomings. To minimize the number of arithmatic operations, several approximations were introduced. Instead of computing the exact contribution of each face for updating the interior boundary points, some simple recipes were employed. This results in communication between the cells that lie on either side of a solid object. That is, the interior boundary point pairs 1 and 2 in Fig. 3. interact, resulting in an erroneous interaction between the inside and outside of the body. It appears that shortcuts may not work, and it may be necessary to consider the exact geometry of the intersecting surfaces when interior boundary conditions are encountered. Since this process is very involved, it may not be acceptable for many problems. Alternative solutions are currently being investigated.

One burden that we have carried over from structured-grid algorithms to unstructured grid is the use of locally one-dimensional approximate Riemann solvers. This conscious introduction of a known problem is due to lack of a better alternative. Several multidimensional Riemann solvers have been considered in the structured-grid world without much success. This problem is accentuated in the case of unstructured grids due to the difficulty in controlling cell shapes. Of course, when sufficiently fine grids are employed, Riemann solvers do not play a major role. But, this doesn't happen in the real world and hence, at least for now, we do have to reckon with errors that arise from locally one-dimensional approximate Riemann solvers.

9. CONCLUSIONS

The most important conclusion that we have arrived at from our experiments with unstructured grid computations is that they offer a chance to prove to the designers that CFD is a tool not just for analysis, and that it may very well be a better alternative to existing design tools such as panel methods. Structured grid solver will still play a major role as an analysis tool and as a tool for understanding some complex flow features.

REFERENCES

[1] V. Shankar, "A Conservative Full Potential, Implicit, Marching Scheme for Supersonic Flows", AIAA Journal, Vol. 20, Nov. 1982, pp. 1508-1514.

[2] S.R. Chakravarthy and K.Y. Szema, "An Euler Solver for Three-Dimensional Supersonic Flows with Subsonic Pockets", AIAA Paper No. 85-0243.

[3] S.R. Chakravarthy, "High Resolution Upwind Formulations for the Navier-Stokes Equations", VKI Lecture Series in Computational Fluid Dynamics, 1988-05.

[4] S. Palaniswamy, S.R. Chakravarthy and D.K. Ota, "Finite-Rate Chemistry for USA-Series Codes: Formulation and Applications", AIAA Paper No. 89-0200, 1989.

[5] Rockwell Science Center CFD Department, "UNIS User Manual", Version 94.11, November 1994.

[6] S.V. Ramakrishnan, C.L. Chen, S.R. Chakravarthy and K.Y. Szema, "Numerical Simulation of Two Opposing High Speed Trains in a Tunnel", AIAA Paper No. 95-0746, 1995.

[7] D.F. Dominik et al., "Navier-Stokes Solution for the Space Shuttle Vehicle using High Fidelity Full Scale Grid Model", AIAA-93-0419, Jan. 1993.

[8] S.R. Chakravarthy, et al., "Computational Fluid Dynamics Capability for Internally Carried Store Separation", Technical Report (Phase III), Contract No. N60530-90-C-0393, Naval Air Warfare Center, Weapons Division, China Lake, CA 93555-6001, May 1994.

[9] A. Harten and S.R. Chakravarthy, "Multi-Dimensional ENO Schemes for General Geometry", UCLA Computational and Applied Mathematics (CAM) Report 91-16.

[10] P.L. Roe, "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", Journal of Computational Physics, Vol. 43, 1981, pp. 357-372.

[11] C. Rowell, V.V. Shankar, W.F. Hall, A.H. Mohammadian, "Algorithmic Aspects and Computing trends in Computational Electromagnetics Using Massively Parallel Architectures", Proceedings of IEEE First International Conference on Algorithms and Architectures for Parallel Processing, Brisbane, Australia, 19-21 April, 1995, Volume 1, editor V. L. Narasimhan.

[12] S.R. Chakravarthy, K.Y. Szema and S.V. Ramakrishnan, "Unification of Exterior and Interior Boundary Conditions for Inviscid Computational Fluid Dynamics", presented at the 5th International Symposium on Computational Fluid Dynamics, Sendai International Center, Japan, Aug. 31-Sep. 3, 1993.

[13] C.L. Chen, K.Y. Szema and S.R. Chakravarthy, "Optimization on Unstructured Grid", AIAA Paper No. 95-0217, 1995.

[14] U.C. Goldberg and S.V. Ramakrishnan, "A Pointwise Version of Baldwin-Barth Turbulence Model", International Journal of Computational Fluid Dynamics, Vol. 1, Dec. 1993.

A Second-Order Finite-Volume Scheme Solving Euler and Navier-Stokes Equations on Unstructured Adaptive Grids with an Implicit Acceleration Procedure

M. Delanaye [*]
Ph. Geuzaine [†]
J.A. Essers [‡]
P. Rogiest [§]

Aerodynamics group, Institute of Mechanics and Aeronautics (C3)
The University of Liege
rue Ernest Solvay 21
B-4000 Liege, Belgium

1. SUMMARY

In this paper, recent advances in the development of a new quadratic reconstruction finite-volume scheme for unstructured polygonal meshes are presented. The scheme is used to discretize the two-dimensional compressible Euler and full Navier-Stokes equations. The quadratic reconstruction is shown to lead to a full second-order accurate discretization of the advective derivatives. The accuracy of the scheme is very weakly dependent on grid distortions, a property which is very attractive for adaptive unstructured grids computations. The pseudo-time integration of the equations is performed by an implicit scheme based on Newton-Krylov techniques. The linear system that arises from the Newton linearization is solved by the GMRES algorithm. The incomplete LU factorization is employed for the system preconditionning. The accuracy, efficiency and robustness of the method are demonstrated on various classical test cases respectively corresponding to inviscid and viscous laminar flows.

2. LIST OF SYMBOLS

∂_ξ	$\frac{\partial}{\partial \xi}$
$\mathbf{s}$	conservative variables
u, Q	any flow variable
$\mathbf{f}$,$\mathbf{g}$	advective fluxes
$\mathbf{F}$,$\mathbf{G}$	viscous fluxes
n_x,n_y	edge normal components
Ω	area of control volume
$\partial\Omega$	contour of control volume
δ_k	length of edge k
$\mathcal{H}$	Hessian matrix
O	cell gravity center
$\mathbf{r}$	position vector
Γ	volume surrounding a cell and bounded by its neighbors
$\Delta_i\odot$	$\odot_i - \odot_O$
QUA	quadratic reconstruction
LIN	linear reconstruction
CON	constant reconstruction
ROE	Roe's flux difference splitting
VL	Van Leer's flux vector splitting
σ	discontinuity detector
Δt	time step
h	local characteristic mesh size

3. INTRODUCTION

During this last decade, many investigations have been carried out to develop efficient numerical techniques for solving the compressible Navier-Stokes equations for complex geometries. Unstructured meshes turn out to be a useful tool to generate grids around general configurations, and offer the powerful capability of adaptation to local flow features. Nevertheless, the inherent distortions present in unstructured grids cause the classical schemes used for structured meshes to be mostly inefficient for computing accurate solutions.

In 1990, Barth and Frederickson [1] proposed the concept of high-order polynomial reconstruction also named k-exact reconstruction schemes. Depending on a discrete set of cell values, a high-order cell-by-cell reconstruction of the flow variables is performed. A high-order Gauss quadrature coupled with an approximate Riemann solver is used to evaluate the flux balance integrals. This approach essentially corresponds to the generalization of the Godunov method to high-order schemes on any type of meshes. It was initially designed for Essentially Non Oscillatory schemes [2,3]. However, the application of the latter to steady state computations proved to be difficult due to the large computational time requirement and to some convergence problems [4]. Barth [5] developed a quadratic reconstruction with a fixed support stencil in the frame of a cell-vertex finite-volume scheme. At the same time, Essers et al. [6,7] proposed a non fully conservative finite-volume scheme for structured meshes which also preserved quadratic polynomials. Despite its lack of conservativity, the accuracy and the robustness of the method were demonstrated by the computations of viscous flows on very distorted meshes.

In the frame of the present research [8-10], we contribute to the work of these authors by developing a robust and accurate scheme. The accuracy of the scheme is determined by the use of an original quadratic reconstruction of the flow variables and a second-order Gauss quadrature within a cell-

[*] Research Assistant
[†] F.R.I.A. Research Assistant
[‡] Professor
[§] F.N.R.S. Research Assistant

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

centered finite-volume solver. The quadratic reconstruction can be interpreted as a higher-order generalization of the robust Green-Gauss linear reconstruction [11,12] widely employed in unstructured solvers. The method is designed to deal with grids that contain general polygonal cells with an arbitrary number of edges. It provides a high flexibility. Hybrid grids which are of high interest when solving viscous flows can be used. Adaptation by h-refinement and coarsening is employed. The second-order accuracy of the scheme with respect to the discretized advective derivatives can be demonstrated, and is achieved regardless of the amount of grid distortions. The viscous term is discretized by using a central approximation. The monotonicity of the solution is guaranteed by using a discontinuity detector that switches the scheme to a constant reconstruction.

For large problems, convergence rates obtained by explicit methods (Runge-Kutta), even with some acceleration techniques such as local time-stepping or residual averaging, finally remain insufficient. The speed-up of the convergence can be achieved by employing a multigrid approach or/and implicit schemes [13,14]. Venkatakrishnan and Barth [15] tested a fully implicit scheme for unstructured meshes, wherein the system arisen from the Newton's linearization was solved by direct methods. However, that attempt showed that direct methods, despite their robustness, are plagued by extremely prohibitive memory and computational requirements. As an alternative, iterative implicit solvers have been studied by many authors [16–18]. The Newton-Krylov methods [19] have turned out to be really successful for a broad class of problems. Within the frame of the Inexact Newton methods [20], iterative solvers based on Krylov subspace generation are employed to approximately solve the linear system that arises from the Newton linearization. Among others, the Generalized Minimum Residual (GMRES) algorithm of Saad and Schultz [21] has proved to be very efficient thanks to its robustness. We employ it in its finite-difference version that has the major advantage not to require the storage of the system Jacobian. To accelerate the convergence of conjugate-gradient like algorithms, preconditionning is highly recommended for clustering the eigenvalues of the matrix [22]. For that purpose, we use the incomplete LU factorization with no fill in.

4. FINITE VOLUME DISCRETIZATION

Consider a finite-volume discretization of the Navier-Stokes equations onto a set of polygonal cells whose number of edges N_i can be arbitrary:

$$\int_{\Omega_i} \partial_t \mathbf{s}\, d\Omega_i + \oint_{\partial\Omega_i} [(\mathbf{f}+\mathbf{F})n_x + (\mathbf{g}+\mathbf{G})n_y]\, d(\partial\Omega_i) = 0 \tag{1}$$

Within the frame of the cell-centered variant of the finite-volume method, we associate to each polygonal cell a set of conservative variables ($\mathbf{s}_i$) which refer to the unknowns at the cell gravity center (node). A second-order spatial discretization of the time derivatives of (1) can therefore be achieved without requiring any artifice such as for example the mass lumping implicitly present in the cell-vertex technique. The Navier-Stokes equations reduce to the following semi-discretized conservative system of the non-linear equations:

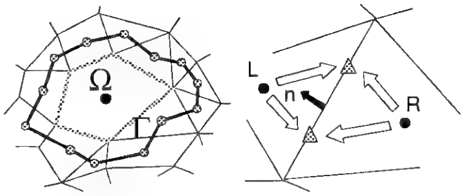


Fig. 1: Control volume, quadrature points

$$\frac{d\mathbf{s}_i}{dt} + \frac{1}{\Omega_i}\sum_{k=1}^{N_i}(\mathbf{f}_n^k + \mathbf{F}_n^k) = 0 \tag{2}$$

$$\text{where : } \mathbf{f}_n^k = \int_{\delta_k} (\mathbf{f} n_x^k + \mathbf{g} n_y^k)\, d\delta_k$$

$$\mathbf{F}_n^k = \int_{\delta_k} (\mathbf{F} n_x^k + \mathbf{G} n_y^k)\, d\delta_k$$

5. ADVECTIVE DERIVATIVES

Obviously the accuracy of the scheme (2) is essentially dependent on the numerical integration of the non-linear advective flux along the mesh edges. Two steps follow:

- First, a reconstruction phase reconstructs the flow variables in the cell from the discrete values at the neighboring cell gravity centers.
- Secondly, a high-order Gauss quadrature integrates the upwind numerical flux computed by a Riemann Solver.

5.1 Preliminary note on the order of accuracy

Various definitions of the accuracy of a scheme exist in the CFD community. In this paper, we use a definition which is usual in the finite-difference community, i.e we refer to the accuracy obtained in the evaluation of the advective and diffusive derivatives appearing in the equations. Hence, second-order accuracy on first order derivatives (like for the advective part of the Euler equations) means that the error on these first order derivatives for any sufficiently smooth function u should decrease quadratically when the mesh is refined similarly in all space directions. The dominant term of the truncation error is therefore proportional to third-order derivatives times the square of a local characteristic mesh size:

$$\partial_\xi u_{num} - \partial_\xi u_{exact} \simeq K\,(\partial^3_{\xi\xi\xi} u)\, h^2 \tag{3}$$

with K a constant.

A scheme discretizing advective derivatives is considered as second-order accurate if it leads to an exact evaluation of the latter when the corresponding flux vectors are any quadratic (or linear) function of the cartesian coordinates of the physical domain. Indeed, in that case, the third and higher order derivatives appearing in the truncation error (3) obviously vanish.

In the literature, the most frequently employed cell reconstruction uses a representation of the flow variables based on linear polynomials. Clearly, it can only evaluate exactly a linear function, but not a quadratic function. This means that the dominant term of the truncation error involves a second order derivative and can be written:

$$\partial_\xi u_{num} - \partial_\xi u_{exact} \simeq K' \left(\partial^2_{\xi\xi} u\right) h \qquad (4)$$

with $K' \neq 0$ a constant.

The resulting scheme is therefore first-order accurate only. However, when the mesh is sufficiently regular, it can be demonstrated that K' is equal to 0, and the second-order accuracy is recovered. Similarly, for irregular meshes, the dominant error on the advective derivative computed by a constant reconstruction involves a first order derivative with a coefficient that does not tend to zero with the mesh size. For these meshes, the constant reconstruction is thus inconsistent, but consistency (i.e. first-order accuracy) is recovered on regular grids.

By extending the linear reconstruction to a quadratic reconstruction defined as a third-order truncated Taylor series expansion of the variables around the cell gravity center, a quadratic function can be reconstructed exactly provided that the numerical gradient and Hessian matrix are respectively computed with a second-order and a first-order accuracy at least :

$$u_{rec}(\mathbf{r}) = u_O + \Delta\mathbf{r}^T \nabla u_O + \tfrac{1}{2}\Delta\mathbf{r}^T \mathcal{H}_O \Delta\mathbf{r} \qquad (5)$$

with $\Delta\mathbf{r} = \mathbf{r} - \mathbf{r}_O$, and u is any flow variable.

Therefore, if the numerical integration of the numerical flux is sufficiently accurate, a second-order accurate scheme is obtained without any assumption about the grid regularity. This property is quite attractive when using unstructured grids which are usually very irregular.

The calculation of the flux $\mathbf{f}^k_n$ (see eq. (2)) through each edge is performed by a high-order numerical integration of the flux functions using the n-points Gauss quadrature:

$$\mathbf{f}^k_n = \delta_k \sum_{j=1}^{n} \omega_j \left[\mathbf{f}(x^k_j, y^k_j) n^k_x + \mathbf{g}(x^k_j, y^k_j) n^k_y\right] \qquad (6)$$

where (x^k_j, y^k_j) are the coordinates of the Gauss quadrature point j, ω_j denotes the weight associated with this point.

By using n quadrature points, the formula (6) allows the exact integration of polynomials with degree $2n - 1$ at most. To meet the second-order accuracy requirement described in section 5.1, *two quadrature points* are at least needed in order to compute exactly the flux integral of a quadratic polynomial of the cartesian coordinates. Schemes employing linear reconstruction only necessitate *one quadrature point* (located at the mid-point edge), but they are usually first-order accurate only as already mentioned above. Essers et al. [7] however proved that a one quadrature point integration can produce a full second-order scheme even for very irregular meshes. This accuracy can only be recovered by applying a non conservative correction to the scheme, which definitely constitutes a drawback with respect to the present method.

A Riemann solver such as the Roe's flux difference splitting or the Van Leer's flux vector splitting is employed to compute the upwind numerical flux at each quadrature point.

5.2 Reconstruction phase

5.2.1 Fixed support stencil

Contrary to the Essentially Non Oscillatory schemes, the stencil that supports the reconstruction is fixed during the iterations. The major advantage is an important saving of CPU time, but an accuracy deterioration occurs in the vicinity of discontinuities. Indeed, to satisfy some monotonicity requirements in shocks or other regions with strong flow gradients, the scheme should locally reduce to a constant reconstruction whatever the methodology employed : TVD [23] , LED [24], or hybrid schemes [25].

5.2.2 A classical linear reconstruction

By dropping off the quadratic terms of (5), a linear reconstruction is obtained which in fact corresponds to the extension of the classical finite-difference Fromm scheme to multiple dimensions:

$$u_{rec}(\mathbf{r}) = u_O + \Delta\mathbf{r}^T \nabla u_O \qquad (7)$$

The gradient at the cell gravity center is computed by the well-known robust Green-Gauss reconstruction widely employed in unstructured grid solvers [5,12].The Green-Gauss theorem is applied to compute the averaged gradient of u over the surface of a bounding control volume:

$$\nabla u_O = \frac{1}{\Gamma} \oint_{\partial\Gamma} u\, \mathbf{n}\, d\partial\Gamma \qquad (8)$$

where Γ denotes a bounding volume surrounding Ω and is defined by the neighbors of Ω (fig. 1). The integral in (8) is discretized by a summation of the contributions of the linear segments of $\partial\Gamma$ obtained from the trapezoidal rule. It leads to a linear combination of the values of the neighboring nodes:

$$\nabla u_O = \mathbf{D}_1 \mathbf{\Delta u} \qquad (9)$$

$$\text{with: } \mathbf{\Delta u} = \begin{bmatrix} u_1 - u_O \\ \vdots \\ u_{N_\Omega} - u_O \end{bmatrix}$$

N_Ω is the number of neighbors of Ω, i.e. the cells connected to Ω by at least a common edge or a common vertex. $\mathbf{D}_1$ is a $2 \times N_\Omega$ matrix with constant coefficients.

5.2.3 Extension of the linear to the quadratic reconstruction

By using a Taylor series expansion of u around node O in (9), the truncation error $\mathbf{E}$ corresponding to formula (9) can be expressed as:

$$\mathbf{E} = \mathbf{Er}\, \nabla^2 u_O \qquad (10)$$

$$\text{with: } \nabla^2 u_O = \begin{bmatrix} \partial^2_{xx} u_O \\ \partial^2_{yy} u_O \\ \partial^2_{xy} u_O \end{bmatrix}$$

Note that $\mathbf{Er}$ is a 2×3 matrix containing constant coefficients of $\mathcal{O}(h)$. For arbitrary meshes, second-order accuracy is nevertheless recovered by subtracting $\mathbf{E}$ from the right-hand-side of (9):

$$\nabla u_O = \mathbf{D}_1 \mathbf{\Delta u} - \mathbf{Er}\, \nabla^2 u_O \qquad (11)$$

This second-order numerical gradient does indeed depend on some sufficiently accurate (first-order at least) second-order derivatives. By replacing (11) in (5), we obtain a quadratic reconstruction for which the only unavailable coefficients are the second-order derivatives of u:

$$\underbrace{u_{rec}(\mathbf{r}) = u_O \overbrace{+\mathbf{\Delta r}^T \mathbf{D}_1 \mathbf{\Delta u}}^{\text{linear part}} + \left[-\mathbf{\Delta r}^T \mathbf{Er} + \left[\tfrac{1}{2}\Delta x^2 \tfrac{1}{2}\Delta y^2 \Delta x \Delta y\right]\right] \nabla^2 u_O}_{\text{quadratic part}} \tag{12}$$

The second order derivatives are computed by a technique sometimes referred to as the minimum-energy reconstruction [5,26]. It simply consists in fitting the cell quadratic polynomial u_{rec} to the values of the neighboring nodes. The following functional is minimized with respect to $\nabla^2 u_O$:

$$\sum_{i=1}^{N_\Omega} \left(u_{rec}(\mathbf{r}_i) - u_i\right)^2 \tag{13}$$

which is equivalent to solve in the least square sense the following linear system of N_Ω equations and 3 unknowns:

$$\left(\mathbf{\Delta}_2 - \mathbf{\Delta}_1 \mathbf{Er}\right) \nabla^2 u_O = \left(\mathbf{I} - \mathbf{\Delta}_1 \mathbf{D}_1\right) \mathbf{\Delta u} \tag{14}$$

with:

$$\mathbf{\Delta}_1 = \begin{bmatrix} \Delta x_1 & \Delta y_1 \\ \vdots & \vdots \\ \Delta x_{N_\Omega} & \Delta y_{N_\Omega} \end{bmatrix} \mathbf{\Delta}_2 = \begin{bmatrix} \frac{1}{2}\Delta x_1^2 & \frac{1}{2}\Delta y_1^2 & \Delta x_1 \Delta y_1 \\ \vdots & \vdots & \vdots \\ \frac{1}{2}\Delta x_{N_\Omega}^2 & \frac{1}{2}\Delta y_{N_\Omega}^2 & \Delta x_{N_\Omega} \Delta y_{N_\Omega} \end{bmatrix}$$

The normal equations are non singular provided that the stencil contains at least 6 nodes. Although this condition is generally fulfilled for interior nodes, it may not be for boundary nodes. In that case, the stencil must be enlarged by incorporating the neighbors of the neighbors sharing an edge with the concerned node. The solution of (14) corresponds to a first-order approximation of the second order derivatives which can be expressed as a linear combination of the nodal values:

$$\nabla^2 u_O = \mathbf{D}_2 \mathbf{\Delta u} \tag{15}$$

with $\mathbf{D}_2$ a $3 \times N_\Omega$ matrix with constant coefficients:

$$\mathbf{D}_2 = \left[\left(\mathbf{\Delta}_2 - \mathbf{\Delta}_1 \mathbf{Er}\right)^T \left(\mathbf{\Delta}_2 - \mathbf{\Delta}_1 \mathbf{Er}\right)\right]^{-1} \left(\mathbf{\Delta}_2 - \mathbf{\Delta}_1 \mathbf{Er}\right)^T \left(\mathbf{I} - \mathbf{\Delta}_1 \mathbf{D}_1\right)$$

All the matrices involved in the reconstruction $\mathbf{Er}$, $\mathbf{D}_1$ and $\mathbf{D}_2$ are preprocessed and stored.

Note that variables u are actually the primitives variables. As a result, the gradients of the primitives variables are therefore directly available for the computations of the viscous fluxes. For inviscid flows, the conservative variables could however be used as well.

5.3 Monotonicity of the reconstruction

High-order schemes produce oscillations in the vicinity of discontinuities. That problem can be overcome in two different ways: either by selecting another stencil for the reconstruction which does not involve the discontinuity (ENO schemes [27]), or by modifying the reconstruction within the same stencil (TVD schemes [23]).

The design of multidimensional limiters has been introduced by Barth and Jespersen [11]. However, as shown by Venkatakrishnan [28], such limiters may severely hamper the convergence to the steady state. This problem is still more dramatic when employing implicit schemes with large CFL numbers. Venkatakrishnan [28] proposed some modifications to the limiter, and obtained convergence at the price of the evaluation of an additional constant.

We employ another approach by using the rather old idea of hybrid schemes [25], however applied to the reconstruction. The quadratic reconstruction is switched to a monotone constant reconstruction in the vicinity of discontinuities. While in ‘‘smooth flows regions’’, it remains unaltered.

This is easily achieved with the formulation (12), but requires a discontinuity detector, that is taken of the form:

$$\sigma_O = \frac{\displaystyle\sum_{i=1}^{N_\Omega} \left|\mathbf{\Delta r}_i (\nabla_i u - \nabla_O u)\right|}{\displaystyle\sum_{i=1}^{N_\Omega} \left[\left(|\mathbf{\Delta r}_i \nabla_i u| + |\mathbf{\Delta r}_i \nabla_O u|\right) + \gamma \left(|u_i| + |u_O|\right)\right]} \tag{16}$$

u is the pressure or/and the velocity norm. The complete form of γ, which acts as a filter term, is given in reference ([9]).

Formula (16) is an extension of the error indicator developed by Lohner [29] for transient finite-element computations.

By construction σ_O is always bounded by 1, and provided a threshold value β, a discrete discontinuity detector σ_O can be defined at each node O:

$$\begin{array}{ll} \text{if} \quad \sigma_O < \beta & \Longrightarrow \quad \sigma_O = 1 \\ \text{if} \quad \sigma_O \geq \beta & \Longrightarrow \quad \sigma_O = 0 \end{array} \tag{17}$$

where β is usually chosen close to 0.2, and turns out to be relatively case independent.

The quadratic reconstruction (5) is finally modified as follows:

$$u_{rec}(\mathbf{r}) = u_O + \sigma_O \left[\mathbf{\Delta r}^T \nabla u_O + \tfrac{1}{2}\mathbf{\Delta r}^T \mathcal{H}_O \mathbf{\Delta r}\right] \tag{18}$$

σ_O is computed once at the beginning of each Newton's iteration. Unfortunately, the detector is sometimes found to fluctuate at some nodes usually located in the neighborhood of the discontinuities. At these locations, it may indeed appear difficult to decide whether the cells involve the discontinuity or not. However, after a sufficient residual decay, the iterative process can be supposed to be close enough to the solution, and the detector is frozen everywhere for the rest of the convergence. A similar but more complex strategy, which has not been tested in this paper yet, can be found in reference ([9]).

6. DIFFUSIVE DERIVATIVES

A centered discretization is used to discretize the viscous terms of the Navier-Stokes equations. The viscous flux is estimated at one quadrature point located at the mid-edge, which requires the evaluation of the gradients of the primitive variables at the nodes. As pointed out in section 5.2.3, these gradients have been previously computed with a second-order accuracy during the quadratic reconstruction phase. They are obtained at the quadrature point by using a linear interpolation between the left (L) and the right (R) neighbors of the edge. Strictly speaking, that procedure is only valid if the mid-edge point lies on the line joining the left and right neighbors. If it does not, the following modified interpolation formula has to be used:

$$\begin{aligned}\partial_x u_P = \; & \alpha\partial_x u_R + (1-\alpha)\partial_x u_L \\ & +\Delta x_{PQ}(\alpha\partial^2_{xx}u_R + (1-\alpha)\partial^2_{xx}u_L) \\ & +\Delta y_{PQ}(\alpha\partial^2_{xy}u_R + (1-\alpha)\partial^2_{xy}u_L)\end{aligned} \tag{19}$$

where P is the quadrature point, Q the projection of P on $L\vec{R}$, and $\alpha = \frac{L\vec{P}\, L\vec{R}}{||L\vec{R}||^2}$

The accuracy of that discretization is restricted to first-order for arbitrary meshes, but remains second-order when the mesh is sufficiently regular. That accuracy limitation is however not too restrictive because the truncation error is multiplied by a usually small factor (i.e. the inverse of the freestream Reynolds number).

7. IMPLICIT INTEGRATION SCHEME

7.1 Newton's method

As our purpose is to compute steady state solutions of the Euler and Navier-Stokes equations, Newton's method could be directly applied to the steady state equations. However, a well-known drawback of the Newton's method is the need for a sufficiently "close initial guess" in order to guarantee convergence. A common approach to bypass that problem is to consider the unsteady equations and to march in time. Because fully implicit schemes are known to be unconditionally stable, the time step Δt is allowed to increase and finally tend to infinity during the time-marching in order to permit quadratic convergence when approaching the solution. The time-marching strategy can also be interpreted as the addition of an extra $\frac{1}{\Delta t}$ term to the diagonal of the Jacobian of the steady equations in order to increase its magnitude, which in fact corresponds to an under-relaxation procedure.

An Euler-backward time-stepping is employed to discretize the time derivative of the equations (1). A Newton-Raphson iterative process is performed at each time step l to find the solution s^{l+1} of the following system of non-linear equations:

$$\mathcal{F}(\mathrm{s}^{l+1}) = \frac{\mathrm{s}^{l+1}-\mathrm{s}^{l}}{\Delta t} + \mathcal{R}(\mathrm{s}^{l+1}) = 0 \tag{20}$$

The operator $\mathcal{R}(Q)$ corresponds to the discretization of the spatial derivatives of the Euler and Navier-Stokes equations described in the previous sections. Finally, the whole iterative process can be summarized in two loops:

$$\left\{\begin{array}{l}\text{For } l = 0, 1, \ldots \text{ until convergence do:} \\ \text{Set } Q^{(0)} = \mathrm{s}^l \\ \left\{\begin{array}{l}\text{For } n = 0, 1, \ldots \text{ until convergence do:} \\ \text{Solve } \mathcal{J}(Q^{(n)})\delta Q^{(n)} = -\mathcal{F}(Q^{(n)}) \\ \text{Set } Q^{(n+1)} = Q^{(n)} + \delta Q^{(n)}\end{array}\right. \\ \text{Update } \mathrm{s}^{l+1} = Q^{(n+1)}\end{array}\right. \tag{21}$$

where $\mathcal{J}(Q) = \frac{1}{\Delta t} + \frac{\partial\mathcal{R}}{\partial Q}$ is the Jacobian of $\mathcal{F}$.

As pointed out by Kuffer[30], deciding when the Newton loop has to be stopped is not easy. A large residual decrease is not always required, which necessitates many inner iterations and then costs a lot of computational time. Except for unsteady flow computations for which equation (20) must be solved accurately, many authors usually limit the number of inner iterations to one $(n = 0)$. The resulting descent direction is in fact usually accurate enough to decrease the residual satisfactorily. As the time step increases to infinity, the iterative time-marching scheme tends to a Newton-Raphson linearization of the steady state equations. Restricted to one inner loop iteration, the iterative process (21) becomes:

$$\left\{\begin{array}{l}\text{For } l = 0, 1, \ldots \text{ until convergence do:} \\ \;\text{Solve } \mathcal{J}(\mathrm{s}^l)\delta \mathrm{s} = -\mathcal{R}(\mathrm{s}^l) \\ \text{Update } \mathrm{s}^{l+1} = \mathrm{s}^l + \delta \mathrm{s}\end{array}\right.$$

7.2 Inexact Newton's method

Most of the computational time required by a Newton algorithm is essentially devoted to the evaluation of the Jacobian $\frac{\partial\mathcal{R}}{\partial Q}$ and to the solution of a linear system. The exact solution of that system is most of the case not justified when the iterate is far from the solution. It seems to be quite reasonable to solve it approximately by using an iterative solver, which of course saves a lot of CPU time with respect to direct methods. Such a method is referred to as an inexact Newton-method. As shown by Dembo et al.[31], the residual on the linear system must however verify the following rule in order to preserve the quadratic convergence:

$$\begin{gathered}\frac{||r_n||}{||\mathcal{F}(Q^{(n)})||} \leq \eta_n \\ \eta_n = \min\{c||\mathcal{F}(Q^{(n)})||^z, \tfrac{1}{2}\} \text{ with } 0 < z \leq 1\end{gathered} \tag{22}$$

where $r_n \equiv \mathcal{J}(Q^{(n)})\delta Q^{(n)} + \mathcal{F}(Q^{(n)})$, $||.||$ denotes any arbitrary norm in R^n, c is a constant and $\mathcal{F}$ is supposed to be suitably scaled. In our code, we use $c = 0.5$ and $z = 0.5$.

The choice of the iterative solver is obviously essential. These solvers can be grouped in two sets: stationary iterative methods (Jacobi, Gauss-Seidel, SOR, ... etc), and the non-stationary iterative methods (Krylov subspace algorithms, Chebychev iteration, ... etc). This last set differs from stationary methods in that the computations involve information that change at each iteration. Iterative solvers such as Jacobi and Gauss-Seidel are attractive because they are simple and easy to vectorize. But their main drawback is that their robustness and convergence are only ensured when the matrix exhibits large diagonal terms. Unfortunately, due to the stiff problems generated by the Euler and Navier-Stokes equations, a large diagonal for the system Jacobian requires a important under-relaxation, which indeed destroys the Newton's quadratic convergence. On the contrary, Krylov subspace methods such as the conjugate gradient are known to solve complex problems in a finite number of iterations. Many algorithms derived from the conjugate gradient have been developed to deal with a broad range of problems. Among others, the Generalized Minimum

Residual (GMRES) of Saad and Schultz [22] is designed to solve non-symmetric linear systems.

7.3 The finite-difference GMRES algorithm

The GMRES is a projection method for solving a linear system

$$Ax = b \tag{23}$$

that seeks an approximate solution x_m from an affine subspace $x_0 + \mathcal{K}_m$ of dimension m by imposing the Petrov-Galerkin condition:

$$b - Ax_m \perp A\mathcal{K}_m \tag{24}$$

x_0 represents an initial guess to the solution. $\mathcal{K}_m$ is the Krylov subspace of dimension m:

$$\mathcal{K}_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \ldots, A^{m-1}r_0\} \tag{25}$$

with $r_0 = b - Ax_0$.

In other words, the GMRES process successively build an approximate solution x_m at each iteration m so that x_m is orthogonal to the previous search directions in the metric of the matrix A, and minimizes the residual r_m in L2 norm. More details about the implementation can be found in reference [32]. In our code, a *block* variant of the basic algorithm with restart is used. The whole problem is indeed considered as a $n \times n$ system of 4×4 block matrices.

Typically, a Krylov solver such as GMRES does not require the calculation of the Jacobian $\mathcal{J}(Q)$ but only necessitates the computation of a matrix vector product:

$$\mathcal{J}(Q)\,p \tag{26}$$

where p denotes any vector.

For nonlinear equations, this action can be approximated by a finite-difference quotient of the form:

$$\mathcal{J}(Q)\,p = \frac{\mathcal{F}(Q + \epsilon p) - \mathcal{F}(Q)}{\epsilon} \tag{27}$$

An analysis of the convergence of what is referred to as *the inexact Newton/finite-difference projection methods* is given by Brown [20]. The interesting feature of equation (27) is that the calculation and the storage of the Jacobian are not required. Indeed, the computation of the jacobians of the advective and diffusive flux may be very complicated, and the exact jacobian of the Roe's flux difference splitting is very expensive to compute. Furthermore, the introduction of turbulence modelling in the frame of future developments will also lead to difficulties for deriving jacobians. The stencil of the quadratic reconstruction usually involves an average of 9 to 13 cells. Therefore, the required storage should amount from 144 to 208 words per cell, which is quite expensive.

A proper choice of the parameter ϵ in (27) is given by the analysis of Dennis and Schnabel [33] :

$$\epsilon\,||p|| = \sqrt{\eta} \tag{28}$$

where η is the machine zero and $||.||$ represents the RMS norm.

7.4 Preconditionning

The convergence of Krylov solvers is very dependent on the eigenvalues of the matrix. To accelerate the convergence, the use of a preconditionner that clusters the eigenvalues to each other is strongly recommended. The preconditionner should be as close as possible to the inverse of the matrix. In practice, it should allow a fast linear system resolution. Preconditionners based on stationary methods (diagonal preconditionner or Jacobi, Gauss Seidel, SOR) have been widely employed. A comparison of different preconditionning techniques can be found in Orkwis et al. [34], and Venkatakrishnan et al. [14]. We use the incomplete LU decomposition [32] (ILU) which has been demonstrated to be a very efficient preconditionning strategy. It is generally employed in its simpler version named $ILU(0)$ for which no fill in is permitted during the LU decomposition. In other words, the non zero elements of the preconditionner are located at the same location as those of the initial matrix. This has the advantage of a fixed and minimum memory requirement. However, this decomposition can turn out to be too weak for stiff problems. We have developed and actually use a $block$ version of the $ILU(0)$.

The choice between right or left preconditionning is of importance. The use of right preconditionning is beneficial. Indeed, when using left preconditionning, all the residual vectors and their norms correspond to preconditionned and thus scaled residuals. Hence, it could be difficult to know whether the algorithm needs to be stopped. On the contrary, right preconditionning allows the use of the actual residuals.

Although the Jacobian matrix does not need to be formed in the GMRES algorithm, an approximate form of it is however still required for the *preconditionning*. The support stencil of the quadratic reconstruction is large, it is therefore prohibitive to take all the neighbors into account. As suggested by many authors, an approximate Jacobian may be computed by using a constant reconstruction which only depends on the edge-neighbors or distance-one neighbors. In order to minimize the bandwith and thus the fill-in of the decomposition, a reverse Cuthill-McKee ordering is performed in a preprocessing step .

In most of the results presented in this paper, the Roe's flux difference splitting is employed. It is quite a complex and expensive task to derive analytically even an approximate form [35] of the jacobian of the latter scheme. One alternative is to use the easily available jacobian of the Van Leer's scheme in the preconditionner, which costs 2 to 3 times less computational time than the Roe's scheme jacobian. A comparison between both preconditonners is addressed in the section devoted to the presentation of the results.

It should be mentioned that up to now the contribution of the viscous flux jacobian is not introduced in the preconditionner.

7.5 Time step increment control

As explained in the previous section, the Newton's method is implemented in a time-stepping form. The evolution in time is monitored by the time step. During the time-marching, the time step is increased to infinity in order to ultimately achieve the Newton's quadratic convergence. Like many authors, this is performed by employing an empirical formula in which the CFL number varies according to the inverse of a residual norm:

$$CFL^{l+1} = CFL_0 \left(\frac{||\mathcal{R}(\mathbf{s}^0)||}{||\mathcal{R}(\mathbf{s}^l)||} \right)^p \tag{29}$$

There indeed subsists two different parameters to tune in order to optimize the convergence rate : the initial CFL number and the exponent p. Typical values of the latter parameters are : $CFL_0 = 10$ and $p = 0.5$.

8. BOUNDARY CONDITIONS

The treatment of the boundary conditions has a strong influence on the convergence of an implicit scheme. For inviscid flow computations, we use a very convenient procedure, wich consists in imposing the boundary conditions in a weak manner via the modification of the advective flux through the boundary edges. Hence, according to the boundary type, some of the flow variables are imposed at the quadrature points of the edges, and others are computed from their values at interior nodes using extrapolation formulas similar to those used to evaluate left and right values at the quadrature points of inner edges. For viscous flow computations, inlet and outlet boundaries are treated in a similar way as inviscid boundary conditions. At the solid walls, the viscous flux is modified in order to impose the noslip boundary condition:

$$\mathbf{F}_n^k = \begin{bmatrix} 0 \\ -\mu \, \mathbf{n} \, \nabla u \\ -\mu \, \mathbf{n} \, \nabla v \\ \mathbf{q} \, \mathbf{n} \end{bmatrix} \delta_k \tag{30}$$

That method however turns out to be generally too weak to correctly satisfy the no slip condition. Two additional procedures have been tested. The first corresponds to the introduction of *dummy nodes* in the stencil of the boundary cells. These dummy nodes are located at the mid-point of boundary edges. The flow variables at these nodes are extrapolated or imposed by the no slip boundary condition before each evaluation of the flow derivatives. That method has been implemented in a fully implicit manner and successfully used for the flat plate boundary layer computation. Unfortunately, the result is not so good for more complex flow computations. For these flows, we have tested another procedure. The boundary nodes are no longer located at the cell gravity center, but at the mid-boundary edge, and the noslip boundary condition is imposed in its strong form at each Newton iteration.

Finally, notice that it is essential to include the contribution of the boundary conditions in the preconditionner. The jacobian of the modified boundary advective flux is calculated analytically for most of the boundary conditions except for the subsonic inlet. For the latter, it is derived from a finite difference formula similar to equation (27).

9. MESH ADAPTATION

The possibility of using a flexible local grid adaptation procedure is a major advantage of unstructured meshes. The objective is to improve the resolution of the flow by successive refinements and coarsenings together with minimizing the number of points involved in the mesh. The edge data structure employed in the code and the relatively insensitivity of the accuracy of the numerical scheme to grid distortions allow the use of very general polygonal cells, and as a result of somewhat distorted meshes. We developed a very general adaptation strategy based on mesh enrichment and coarsening. The method is based on an error indicator of the form (16). Cells whose error indicator lies above a preset threshold are candidates for refinement, while others whose error indicator lies under another preset value are to be possibly coarsened. The refinement strategy, which is implemented for any type of polygons is described in reference ([9]). In particular, triangles and quadrangles can be divided anisotropically depending on the value of an anisotropy sensor based on some standard deviations of the gradients of a flow parameter computed in the directions pointing to the different neighbors of the cell. Two types of coarsening procedures are considered. The first one is based on the refinement history. A tree containing the information between successive meshes is updated during the refinements. It is then rather easy to delete ''son'' cells and to recover the ''parent''. The second method is more general and coarsens the grid by deleting vertices and recombining others to build larger polygons.

10. RESULTS

10.1 Subsonic sine-bump

The effect of the various reconstructions (*quadratic - linear - constant*) has first been tested by computing the inviscid subsonic flow ($M_\infty = 0.5$) in a channel perturbed by a sine bump with a mesh of 1294 cells (fig. 2a). The geometry is defined as follows:

Lower wall:

$$\begin{array}{llll} -0.7 & < x < & 0 & : y = 0 \\ 0 & < x < & 1 & : y = 0.05[1 + sin(2\pi x - \frac{\pi}{2})] \\ 1 & < x < & 1.7 & : y = 0 \end{array}$$

Upper wall: $-0.7 < x < 1.7 : y = 0.7$

The Roe's scheme is employed as Riemann solver. The solutions have been computed for an infinite value of the CFL number and a maximum number of GMRES iterations equal to 60 with a restart every 30 iterations. Figures 2c and 2d show the evolution of the Mach number and the total pressure on the lower wall. The quadratic reconstruction clearly appears to lead to the lowest spurious entropy generation (fig. 2d). Hence, it predicts the highest peak Mach number : 0.835. For the sake of comparison, the peak values respectively calculated with the linear and the constant reconstructions are equal to 0.804 and 0.754. When compared to other reconstructions (results not shown), the symmetry of the solution obtained with the quadratic scheme is almost perfect as can be seen from the iso-mach lines pattern (fig. 2b).

Fig. 2e illustrates the dramatic convergence improvement obtained with the implicit scheme with respect to an explicit integration using a 3 steps Runge-Kutta algorithm. The preconditionner based on the approximate Roe flux difference splitting yields the fastest convergence in terms of CPU time and number of Newton iterations (fig. 2f). The quadratic reconstruction however takes about 25 % more CPU time than the linear reconstruction scheme to achieve the same residual decay.

10.2 Subcritical NACA0012 airfoil

The second test case again illustrates the accuracy gain obtained with the quadratic reconstruction scheme. The inviscid subsonic flow over the NACA0012 airfoil has been computed at a freestream Mach number of 0.63 and an incidence of 2 deg. The mesh contains 4537 cells (fig. 3a). The far-field boundary is located at a distance of 20 chords away from the airfoil. The starting solution corresponds to the uniform flow. The solutions computed with the various reconstructions behave similarly on the lower wall (fig. 3b). However, larger discrepancies occur at the upper wall due to the strong flow acceleration. The highest peak Mach number (0.981) is again obtained with the quadratic reconstruction and agrees very well with the value computed by Paillere (0.983) [36] on the same mesh with a fluctuation splitting scheme. Figure 3d shows the evolution of the total pressure along the wall. Notice that the level of spurious entropy generated by the quadratic reconstruction is very low. The lift coefficient $Cl = 0.323$ compares well with the value computed by Paillere [36] ($Cl = 0.322$), and the purely numerical pressure drag coefficient is found very low, $Cd = 0.00034$. The lift coefficient is however slightly lower than the exact one predicted by a full potential method ($Cl = 0.334$). That difference can be explained by the fact that no vortex correction is imposed at the far-field boundary condition [36]. The error between the present value and the exact one is equal to 3.4 %. According to the work of Thomas and Salas [37], a computation with a mesh of about 20 chords and with no vortex correction should underpredict the lift coefficient with a factor of 4 %.

The influence of the exponent (p) of the CFL update formula (29) on the convergence has been tested (fig. 3e and 3f). The code diverges when the computation is initiated with an infinite CFL number. The convergence history obtained when the GMRES is replaced by an SOR iterative solver is provided in fig. 3f and 3e. Figure 3f clearly shows that the Newton's quadratic convergence is never reached with the SOR strategy. Nevertheless, this strategy turns out to be competitive in terms of the computational cost (fig. 3e).

10.3 Transonic flow over a circular arc bump

To test the accuracy and the performance of the scheme to calculate flows with shock waves, the code is applied to a classical test case: the inviscid transonic flow ($M_\infty = 0.85$) over a circular arc bump in a channel. The use of implicit Newton-Krylov techniques for flows with discontinuities remains difficult because of the modifications of the reconstruction that are required to preserve the monotonicity of the solutions. As explained in section 5.3, the quadratic reconstruction (18) modified by the discontinuity detector is employed to achieve monotone solutions. For this test case, the Van Leer's flux vector splitting is used. The computation is started from a solution previously computed with the constant reconstruction scheme. Indeed, one of the major reported disadvantages of implicit-Newton methods is the time required by the shocks to migrate to their right location. During that phase, the residual actually stagnates. That prevents the CFL number to increase to infinity, and therefore dramatically slows down the convergence. We actually use the two following remedies: the starting solution is obtained with a cheap low order scheme, and we use a grid sequencing strategy with mesh adaptation. The initial mesh contains 1420 rectangular cells (fig. 4a). After three adaptation, the final grid (fig. 4b) involves a lower number of cells (1296), which are very general polygons. The total computational time (not shown here) amounts to 400 CPU sec. on a HP9000/730 workstation (infinite CFL number). Fig. 4c shows the points where the detector automatically activates. In order to avoid endless switches of the latter, it is frozen after 5 Newton's iterations. As can be shown of fig. 4d and 4e, a very crisp shock is captured. Different convergence histories for computations performed on the initial mesh are presented in fig. 4g and 4h. The fastest convergence is again obtained with an infinite CFL number. Those figures also show that an exponent p equal to 2 yields a similar convergence history. For the sake of comparison, the GMRES algorithm appears to be about 3 times faster than the SOR scheme in terms of the computational time.

10.4 Inviscid hypersonic flow over a double-ellipse

We now consider the inviscid flow over the double-ellipse test case proposed in the workshop of Antibes [38] at 30 deg. angle of attack and a Mach number of 8.15. The initial mesh of 2412 triangles (fig. 5a) is adapted three times (9527 cells, fig. 5b). The iso-mach lines pattern is presented in fig. 5c. Nearby, the fig. 5d shows the nodes where discontinuities are automatically detected. Convergence histories are presented for the computation on the final adapted mesh. Notice in fig. 5e the dramatic convergence improvement obtained with implicit scheme with respect to a 4 steps explicit Runge-Kutta scheme. Fig. 5h, 5g and 5i respectively give the evolution of the Mach number, the pressure coefficient and the total pressure along the windward and leeward sides. Our results are compared with those obtained by Gustafsson et al. and Khalfallah et al. published in the workshop proceedings [38]. The pressure coefficient and the Mach number agree with the results of the latter authors. Notice the fair agreement between the computed total pressure and the exact one which can be obtained from the normal shock theory (less than 0.02 % error on the leeward side).

10.5 Supersonic flow around a NACA0012 airfoil

The supersonic flow over the NACA0012 airfoil (Mach = 1.2, angle of incidence = 0 degree) illustrates the flexibility of the adaptation technique and the preservation of the accuracy of the scheme even on very distorted grids. The calculation is started on a triangular mesh shown in fig. 6a. Three adaptations are performed. The final mesh is made of polygons with a number of edges varying from 3 to 7 (fig. 6b and 6c). The detached shock and the oblique shocks attached to the trailing edge are well captured (fig. 6d). The distribution of the Mach number on the upstream and downstream parts of the x-axis as well as along the airfoil is presented in fig. 6e. The present calculation is computed with a 4 steps explicit Runge-Kutta scheme. Up to now, no attempt was made to use the implicit scheme for this test case.

10.6 Laminar viscous flow over a flat plate

The accuracy of the Navier-Stokes code has been assessed by investigating the development of a laminar compressible boundary layer over an adiabatic flat plate. For that calculation, the Mach and the Prandtl numbers are respectively taken equal to 0.5 and 1. The viscosity is proportional to the temperature (Crocco's viscosity law) in order to compare our results with the exact solution predicted by the boundary layer theory. The computation is performed with a full quadratic reconstruction and the Roe's flux difference splitting. The initial CFL number is equal to 10 and the exponent p is 0.5. We noticed quite an important numerical influence of the downstream boundary condition (inviscid subsonic outlet with pressure imposed), that forced us to locate that boundary quite far from the leading edge of the plate, i.e. at a Reynolds number based on x equal to 10,000. The mesh contains rectangular cells. There is an average of 15 cells in the displacement thickness of the boundary layer. An excellent agreement is found between the computed and exact velocity and temperature profiles (fig. 7c). Fig. 7b shows the evolution of the skin friction coefficient along the plate, which also agrees very well with the exact one. Notice also the good agreement, especially in outer part of the boundary layer, between the computed and exact shear stress (fig. 7d). Unfortunately, some deviation occurs near the wall. Our latest investigations show that it seems to be caused by a perturbation coming from the downstream boundary condition. The problem must be further studied. The convergence history is reported in fig. 7a. The relatively slow convergence is attributed on one hand to the fact that no contribution of the viscous terms jacobians is introduced in the preconditionner and on the other hand to the weakness of the $ILU(0)$ decomposition.

10.7 Laminar viscous flow over the NACA0012 airfoil

In this final test case, we consider the laminar flow over a NACA0012 airfoil at 0 deg. incidence with a freestream Mach number of 0.5 and a Reynolds number of 5000. The wall is adiabatic. The Sutherland viscosity law is employed and the Prandtl number is equal to 0.72. The flexibility of the method is illustrated by employing a hybrid grid (fig. 8a). It consists in a structured C-type part around the airfoil and in the wake surrounded by a triangular mesh. The far-field boundary is located at a distance of 33 chords from the airfoil. The cell aspect ratio varies from 100 near the wall to 50,000 in the wake. The iso-mach lines pattern presented in fig. 8b shows the development of the boundary layer and its separation near the trailing edge to form a small recirculation bubble. The pressure and skin friction coefficients are presented in fig. 8c and 8d. Accuracy estimates of the results may be carried out by comparing the location of the separation point (in percents of the chord) and the magnitudes of the pressure and viscous drag coefficients. We obtain $x_{sep} = 81.7\%$, $Cd_p = 0.0227, Cd_v = 0.0320$. These results agree with the reference values obtained by Swanson and Turkel [39] on a 518 × 128 structured mesh ($x_{sep} = 81.4\%$, $Cd_p = 0.02235$, $Cd_v = 0.03299$). Notice however that the present mesh only involves 7709 cells and is relatively coarse in the leading edge region which is responsible for a slightly underprediction of the skin friction. We obtain a maximum peak value of 0.143 instead of the reference value 0.15. Moreover, the cell longitudinal dimension in the region of the separation point is also relatively large: about 1 % of the chord.

11. CONCLUSION

In this paper, an original quadratic reconstruction finite-volume scheme for solving the Euler and full Navier-Stokes equations has been presented. The quadratic reconstruction is a higher-order extension of the robust Green-Gauss linear reconstruction. The accuracy of the resulting discretized advective derivatives is second-order, and is insensitive to grid distortions. The robustness and the high accuracy of the scheme have been demonstrated by various computations on very distorted meshes. The Newton-Krylov method based on the GMRES iterative solver has been successfully used to dramatically improve the convergence to steady state with respect to explicit methods. The implicit scheme has been tested on fully subsonic, transonic, supersonic inviscid flows, and on laminar viscous flows computations. For transonic and supersonic flows, a discrete discontinuity detector is employed to switch the scheme to a monotone constant reconstruction. This alternative does not encounter the major problems of the classical multidimensional limiters to drive the convergence to machine accuracy. For inviscid flow test cases when the Roe's flux difference splitting is employed, the preconditionner based on an approximate jacobian of the Roe's flux difference splitting always lead to a faster convergence than a preconditionner based on the Van Leer's flux vector splitting although much cheaper to compute. For viscous flow computations, the ability of the scheme to deal with hybrid grids is a real advantage. The quadratic reconstruction has led to very accurate solutions. However, the proper imposition of the boundary conditions remains a problem. Two methods have been tested. The first one which modifies the stencil of the reconstruction for boundary cells to include the effect of the boundary conditions has been successfully applied for a flat plate boundary layer computation. But, another procedure was required for the computation of a laminar viscous flow around the NACA0012 airfoil. It consists in locating the boundary nodes on the boundary edges rather than at the cell gravity center and then to apply the boundary conditions in their strong form. This modified strategy, which is explicit, unfortunately artificially perturbs the convergence for nodes near solid walls. More efforts should also be devoted to the improvement of the preconditionner which seems to be too weak for viscous flow computations.

12. ACKNOWLEDGMENTS

The works of Ph. Geuzaine and P. Rogiest are presently supported by fellowships awarded by the Fund for the Formation in Research in Industry and Agriculture (F.R.I.A.), and by the National Fund for Scientific Research (F.N.R.S.), respectively. The authors wish to thank Prof. H. Deconinck from Von Karman Institute (Belgium) for providing the mesh employed for the computation of the subsonic flow around the NACA0012 airfoil.

REFERENCES

1. T.J. Barth and P.O. Frederickson. "Higher Order Solution of the Euler Equations on Unstructured Grids using Quadratic Reconstruction". *AIAA paper 90-0013*, 1990.
2. P. Vankeirsbilck. *"Algorithmic developments for the solution of hyperbolic conservation laws on adaptive unstructured grids (Applications to the Euler Equations)"*. PhD thesis, Katholiek Universiteit van Leuven (Belgium) and Von Karman Institute, 1993.
3. R. Abgrall and F.C. Lafon. "ENO schemes on Unstructured Meshes". *VKI Lecture Series 1993-04*, March 1993.
4. A.G. Godfrey, C.R. Mitchell, and R.W. Walters. "Practical Aspects of Spatially High Accurate Methods". *AIAA paper 92-0054*, 1992.
5. T.J. Barth. "Recent Developments in High Order k-Exact Reconstruction on Unstructured Meshes". *AIAA paper 93-0668*, 1993.
6. J.A. Essers, M. Delanaye, and P. Rogiest. "An Upwind-Biased Finite-Volume Technique Solving Compressible Navier-Stokes Equations on Irregular Meshes. Applications to Supersonic Blunt-Body Flows and Shock-Boundary Layer Interactions.". *AIAA paper 93-3377*, 1993.
7. J.A. Essers, M. Delanaye, and P. Rogiest. "An Upwind-Biased Finite-Volume Technique Solving Compressible Navier-Stokes Equations on Irregular Meshes.". *AIAA Journal*, 33(5), 1995.
8. M. Delanaye and J.A. Essers. "Finite Volume with Quadratic Reconstruction on Unstructured Adaptive Meshes Applied to Turbomachinery Flows". *1995 ASME IGTI Gas Turbine Conference, Houston*, June 1995.
9. M. Delanaye and J.A. Essers. "An Accurate Finite-Volume Scheme for Euler and Navier-Stokes Equations on Unstructured Grids ". *AIAA paper 95-1710, 12th CFD Conference, San Diego*, June 1995.
10. M. Delanaye, J.A. Essers, and Geuzaine Ph. "Euler and Navier-Stokes Calculations with a Quadratic Reconstruction Finite Volume Scheme on Flexible Unstructured Grids". *Sixth International Symposium on CFD, Lake Tahoe, Nevada*, September 1995.
11. T.J. Barth and D.C. Jespersen. "The Design and Application of Upwind Schemes on Unstructured Meshes". *AIAA paper 89-0366*, January 1989.
12. D. De Zeeuw and K.G. Powell. "An Adaptively Refined Cartesian Mesh Solver for the Euler Equations". *J. of Comp. Phys.*, 104:56–68, 1993.
13. D.J. Mavriplis and A. Jameson. "Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes". *AIAA Journal*, 28(8):1415–1425, 1990.
14. V. Venkatakrishnan and D.J. Mavriplis. "Implicit Solvers for Unstructured Meshes". *J. of Comp. Phys.*, (105):83–91, 1993.
15. V. Venkatakrishnan and Barth T.J. "Application of Direct Solvers to Unstructured Meshes for the Euler and Navier-Stokes Equations Using Upwing Schemes". *AIAA paper 89-0364, 27th Aerospace Sciences Meeting, Reno, Nevada*, 1989.
16. Wigton L.B., N.J. Yu, and D.P. Young. "GMRES Acceleration of Computational Fluid Dynamics Codes". *AIAA paper 85-1494*, 1985.
17. Z. Johan, T.J.R. Hughes, and F. Shakib. "A Globally Convergent Matrix-free Algorithm for Implicit Time-marching Schemes Arising in Finite Element Analysis in Fluids". *Comp. Meth. in App. Mec. and Eng.*, 87, 1991.
18. D.L. Whitaker. "Three Dimensional Unstructured Grid Euler Computations Using a Fully-implicit, Upwind Methods". *AIAA paper 93-3357, 11th CFD Conference, Orlando*, 1993.
19. P.N. Brown and Saad Y. "Hybrid Krylov Methods for Nonlinear Systems of Equations". *SIAM J. Sci. Stat. Comp.*, 11(3), 1990.
20. P.N. Brown. "A Local Convergence Theory for Combined Inexact-Newton /Finite-Difference Projection Methods". *SIAM J. Num. Anal.*, 24(2), 1987.
21. Y. Saad and M.H. Schultz. "GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems". *SIAM J. Sci. Stat. Comp.*, 7, 1986.
22. Y. Saad. "Preconditioning techniques for nonsymmetric and indefinite linear systems". *J. of Comp. and App. Math.*, 24, 1988.
23. A. Harten. "High Resolution Schemes for Hyperbolic Conservation Laws". *J. of Comp. Phys.*, 49(3):357–393, 1983.
24. S. Tatsumi, L. Martinelli, and A. Jameson. "Flux-Limited Schemes for the Compressible Navier-Stokes Equations". *AIAA Journal*, 33(2):252–261, 1995.
25. A. Harten and G. Zwas. "Self Adjusting Hybrid Schemes for Shock Computations". *J. of Comp. Phys.*, 9(3):368–583, 1972.
26. W.J. Coirier and K.G. Powell. "An Accuracy Assesment of Cartesian-Mesh Approaches for the Euler Equations". *J. of Comp. Phys.*, 117:121–131, 1995.
27. A. Harten and S.R. Chakravarthy. "Multidimensional ENO Schemes for General Geometries". Technical Report No. 91-76, ICASE, 1991.
28. V. Venkatakrishnan. "On the Accuracy of Limiters and Convergence to Steady State Solutions". *AIAA paper 93-0880*, 1993.
29. R. Löhner. "An Adaptive Finite Element Scheme for Transient Problems in CFD". *Comput. Meth. Appl. and Mech. Engrg.*, 61:323–338, 1987.
30. Jürg Küffer. "Fast Implicit Solvers for the Incompressible Navier-Stokes Equations". *Proceedings of Computational Fuild Dynamics '92 Conference , Brussels*, 1:407–412, 1992.
31. R.S. Dembo, S.C. Eisenstat, and T. Steihaug. "Inexact Newton method". *SIAM J. Num. Anal.*, 19(2), April 1982.
32. Y. Saad. "Krylov Subspace Techniques, Conjugate Gradients, Preconditionning and Sparse Matrix Solvers". *VKI Lectures Series 1994-05*, March 1994.
33. J.E. Dennis and R.B. Schnabel. *"Numerical Methods for Unconstrained Optimizations and Non Linear Equations"*. Prentice-Hall, 1983.
34. P.D. Orkwis and J.H. George. "A Comparison of CGS Preconditionning Methods for Newton's method solvers". *AIAA paper 93-3327, 11th AIAA CFD Conference, Orlando*, 1993.
35. T.J. Barth. "Analysis of Implicit Local Linearization Techniques for Upwind and TVD Algorithms". *AIAA paper 87-0595*, 1987.
36. H. Paillère. *"Multidimensional Upwind Residual Distribution Schemes for the Euler and Navier-Stokes Equations on Unstructured Grids"*. PhD thesis, Université Libre de Bruxelles (Belgium) and Von Karman Institute, 1995.
37. Thomas J.L. and M.D. Salas. "Far-Field Boundary Conditions for Transonic Lifting Solutions to the Euler Equations". *AIAA Journal*, 24:1074–1080, 1986.
38. J.-A. Désidéri, R. Glowinski, and J. (Eds) Périaux. *"Hypersonic Flows for Reentry Problems"*, volume 2. Springer Verlag, 1991.
39. R.C. Swanson and E. Turkel. "Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations". *AIAA paper 87-1107*, 1987.

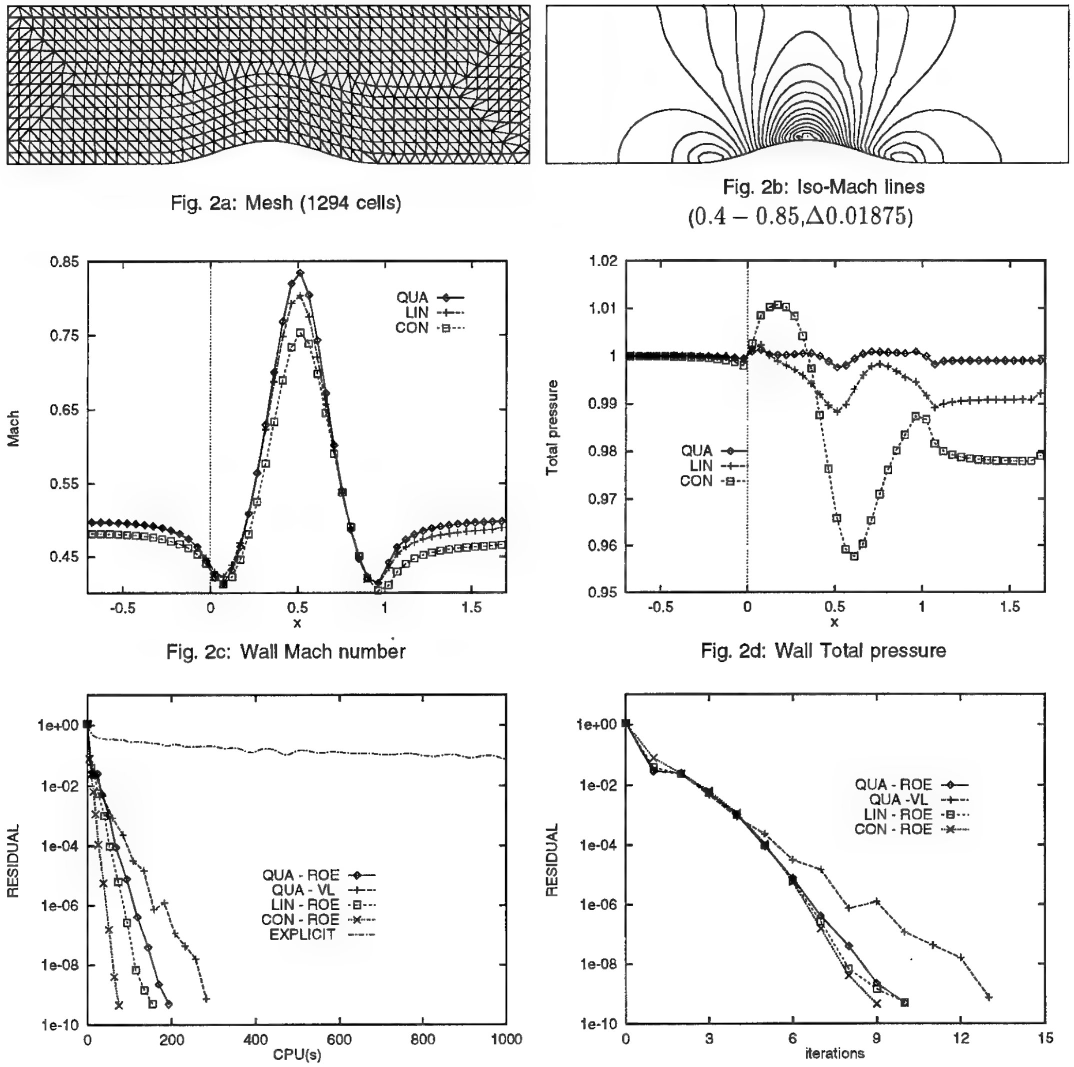


Fig. 2a: Mesh (1294 cells)

Fig. 2b: Iso-Mach lines
$(0.4 - 0.85, \Delta 0.01875)$

Fig. 2c: Wall Mach number

Fig. 2d: Wall Total pressure

Fig. 2e: Convergence - CPU (HP9000)

Fig. 2f: Convergence - Newton iter.

Subsonic sine-bump $M_\infty = 0.5, CFL = \infty$

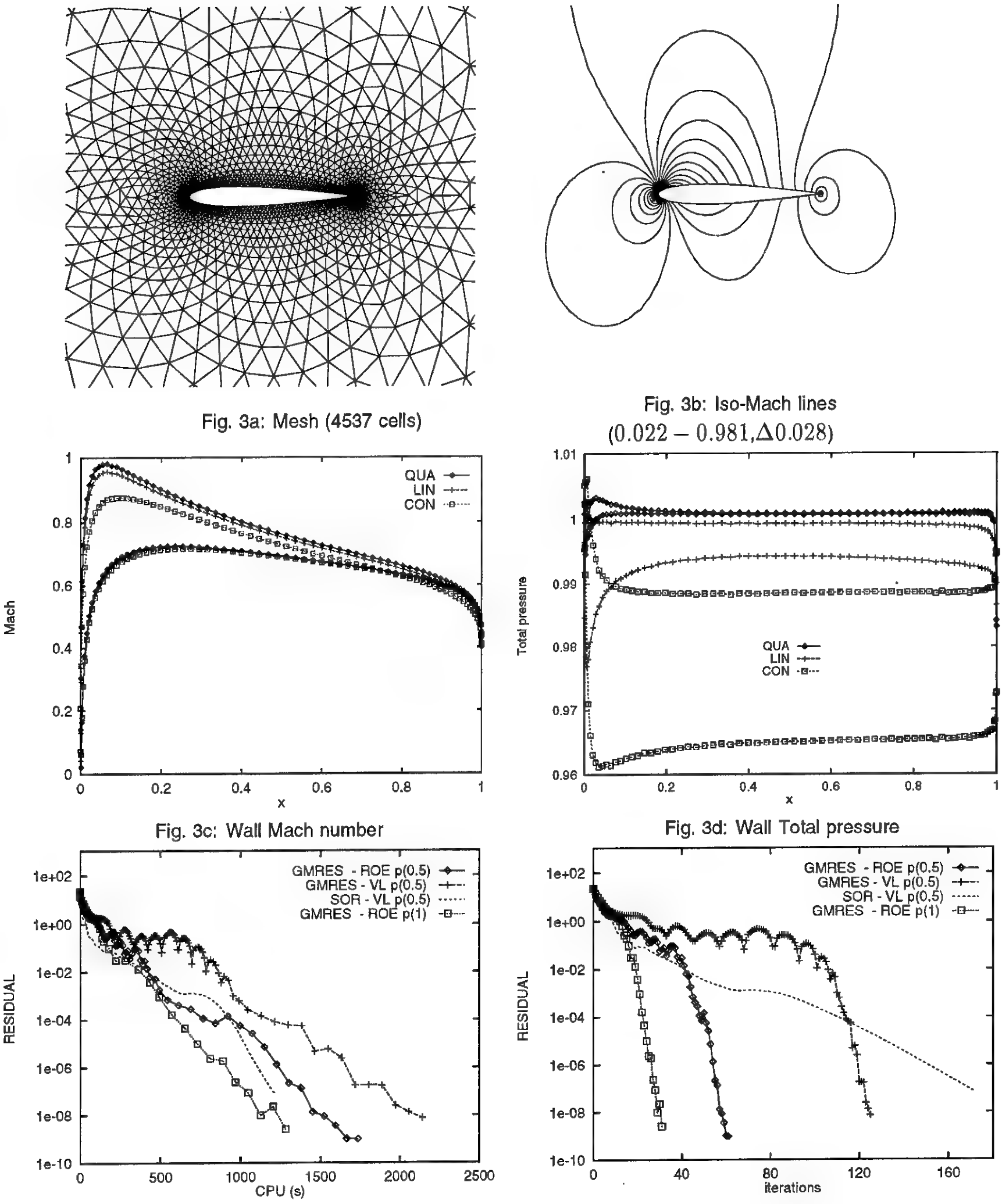


Fig. 3a: Mesh (4537 cells)

Fig. 3b: Iso-Mach lines $(0.022 - 0.981, \Delta 0.028)$

Fig. 3c: Wall Mach number

Fig. 3d: Wall Total pressure

Fig. 3e: Convergence - CPU (DECalpha 250)

Fig. 3f: Convergence - Newton iter.

Subcritical NACA0012 airfoil $M_\infty = 0.63$, incidence = 2 deg., $CFL_0 = 10$

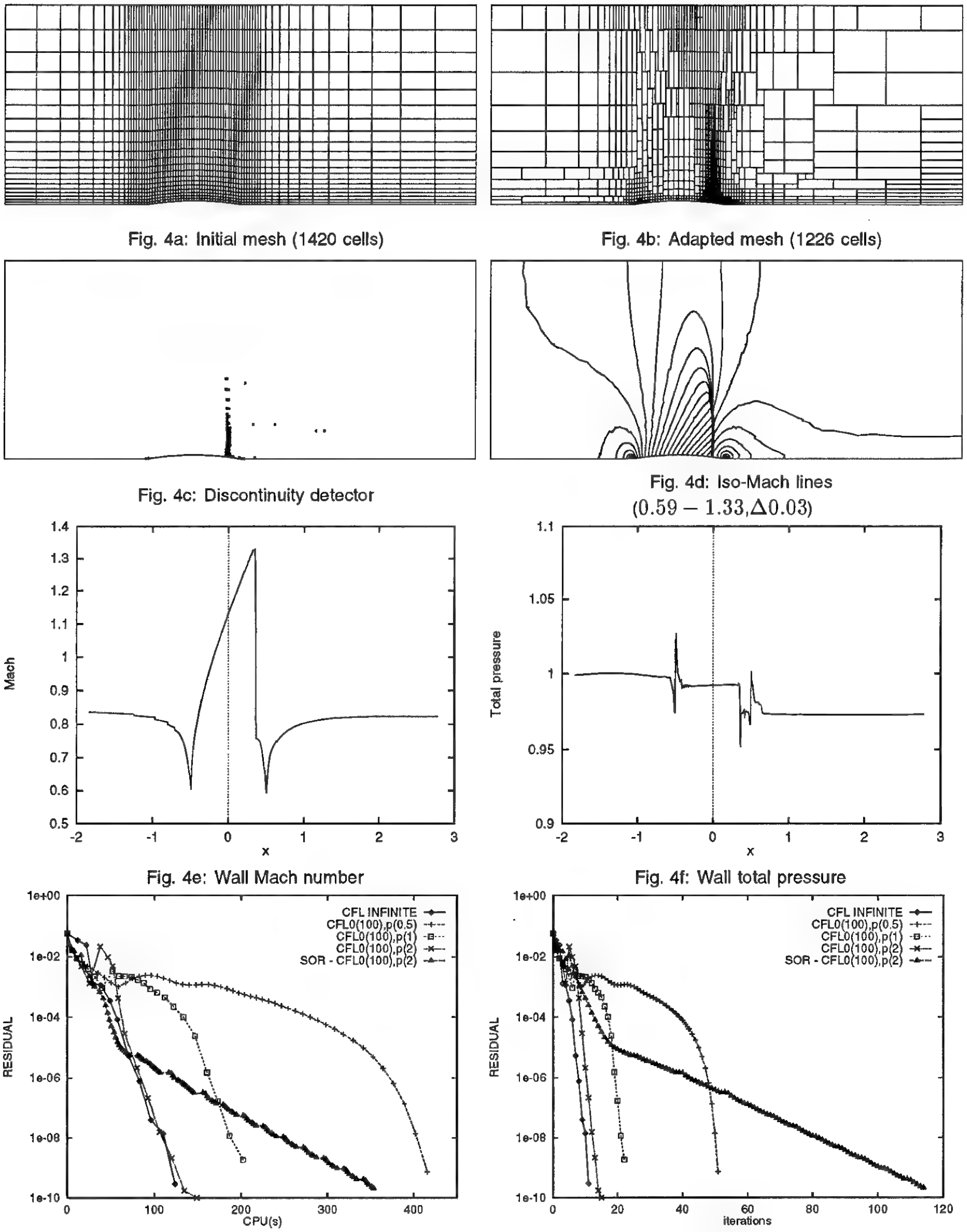


Fig. 4a: Initial mesh (1420 cells)

Fig. 4b: Adapted mesh (1226 cells)

Fig. 4c: Discontinuity detector

Fig. 4d: Iso-Mach lines $(0.59 - 1.33,\Delta 0.03)$

Fig. 4e: Wall Mach number

Fig. 4f: Wall total pressure

Fig. 4g: Convergence - CPU (HP9000)

Fig. 4h: Convergence - Newton iter.

Transonic bump, $M_\infty = 0.85$

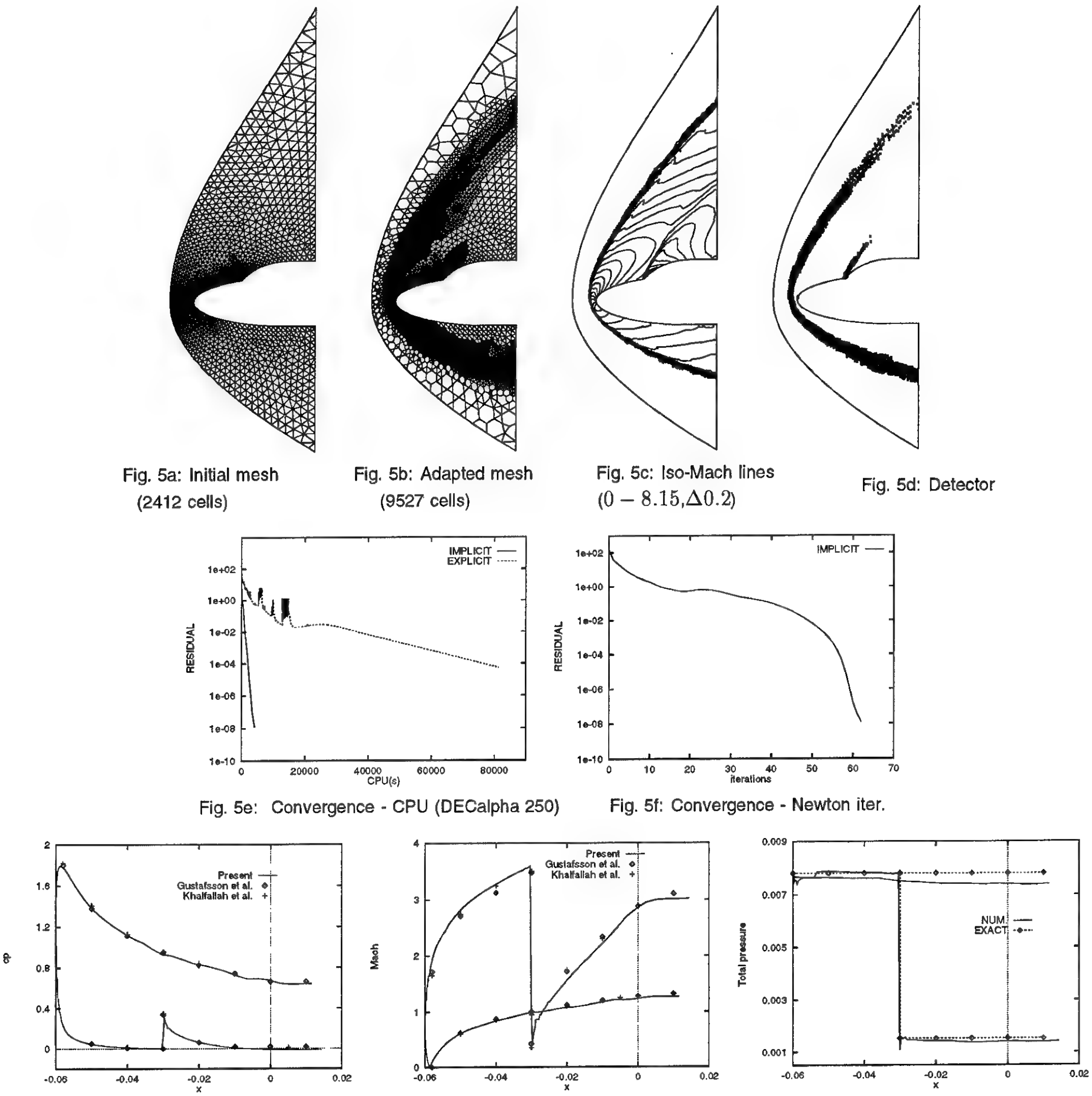


Fig. 5a: Initial mesh (2412 cells)

Fig. 5b: Adapted mesh (9527 cells)

Fig. 5c: Iso-Mach lines $(0 - 8.15, \Delta 0.2)$

Fig. 5d: Detector

Fig. 5e: Convergence - CPU (DECalpha 250)

Fig. 5f: Convergence - Newton iter.

Fig. 5g: Wall Cp

Fig. 5h: Wall Mach number

Fig. 5i: Wall Total pressure

Inviscid double ellipse, $M_\infty = 8.15$, angle of attack = 30 deg.

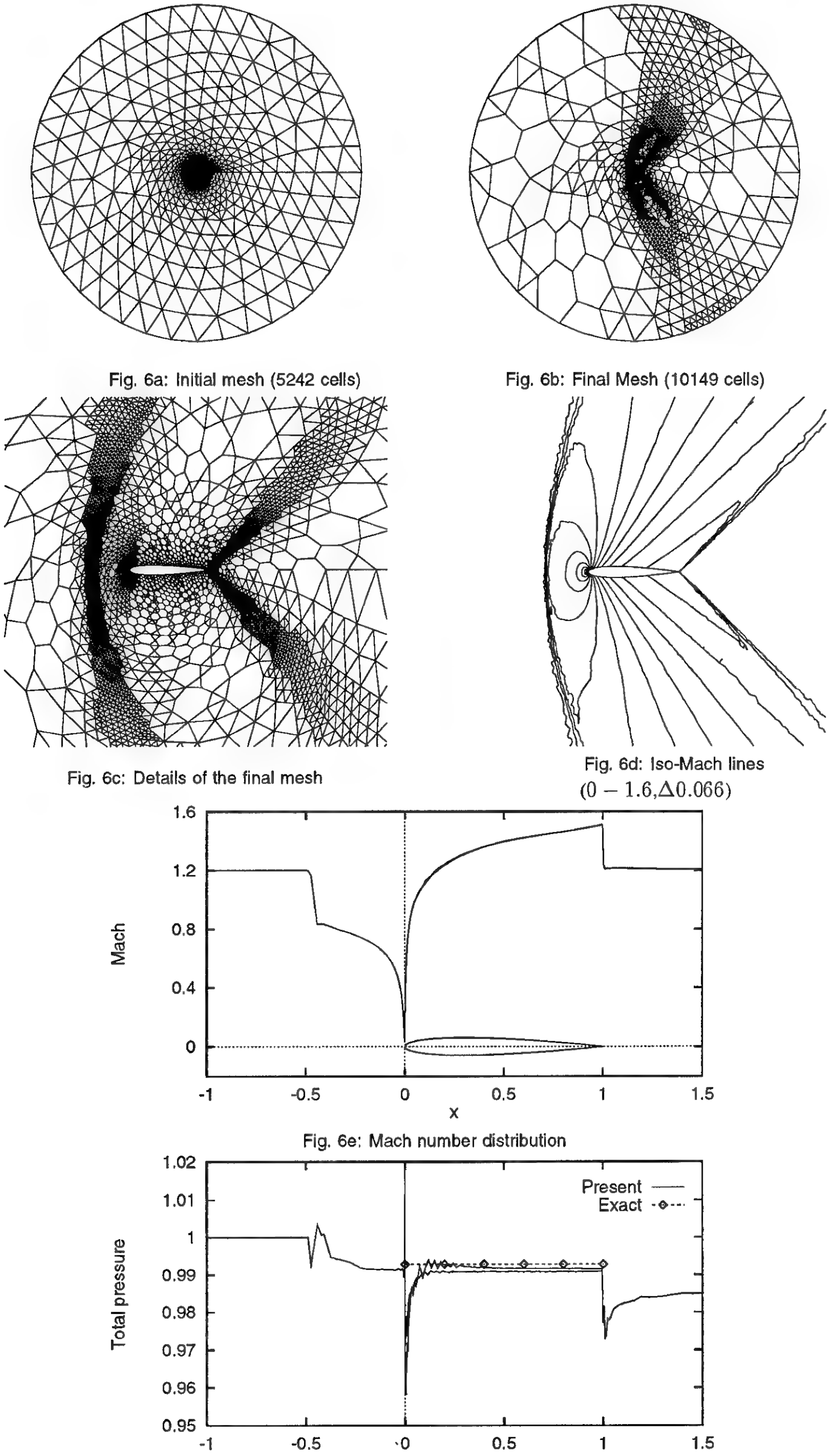


Fig. 6a: Initial mesh (5242 cells)

Fig. 6b: Final Mesh (10149 cells)

Fig. 6c: Details of the final mesh

Fig. 6d: Iso-Mach lines
$(0-1.6, \Delta 0.066)$

Fig. 6e: Mach number distribution

Fig. 6f: Total pressure distribution

NACA0012 airfoil, $M_\infty = 1.2$, angle of attack=0 deg.

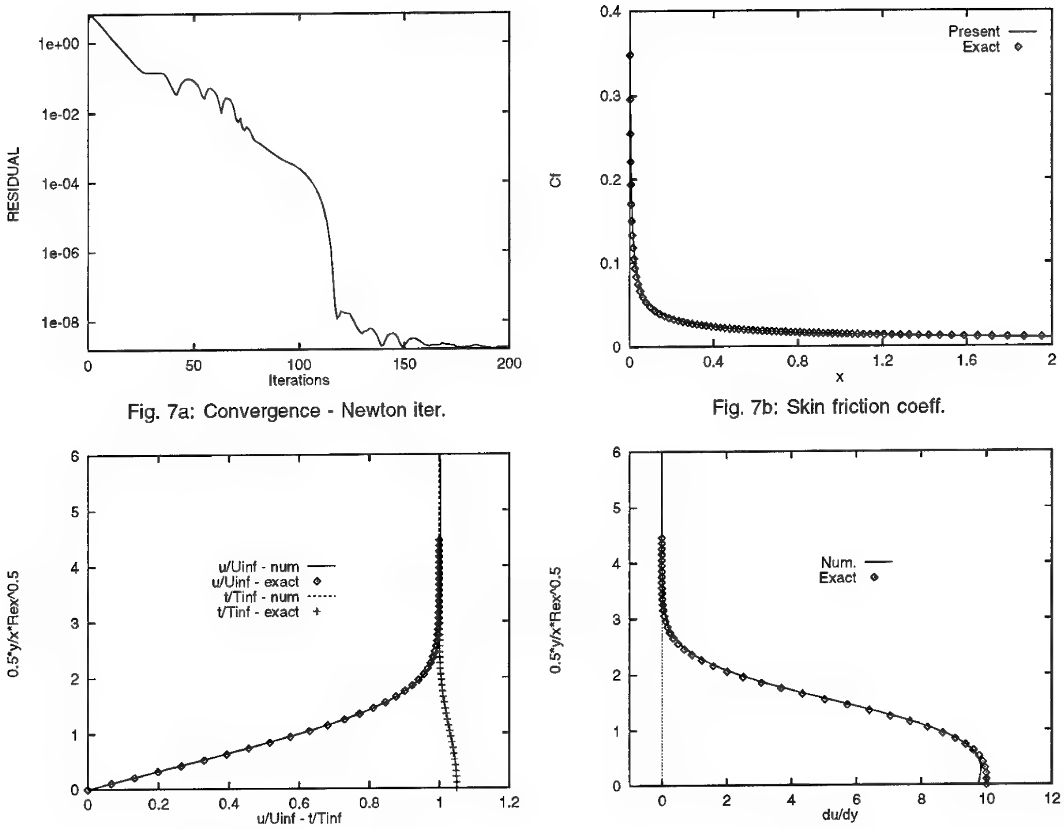


Fig. 7a: Convergence - Newton iter.

Fig. 7b: Skin friction coeff.

Fig. 7c: Velocity and temperature

Fig. 7d: $\partial_y u$

Boundary layer profiles at $Re_x = 4000$

Adiabatic flat plate $M_\infty = 0.5$, $Re_\infty = 2000$

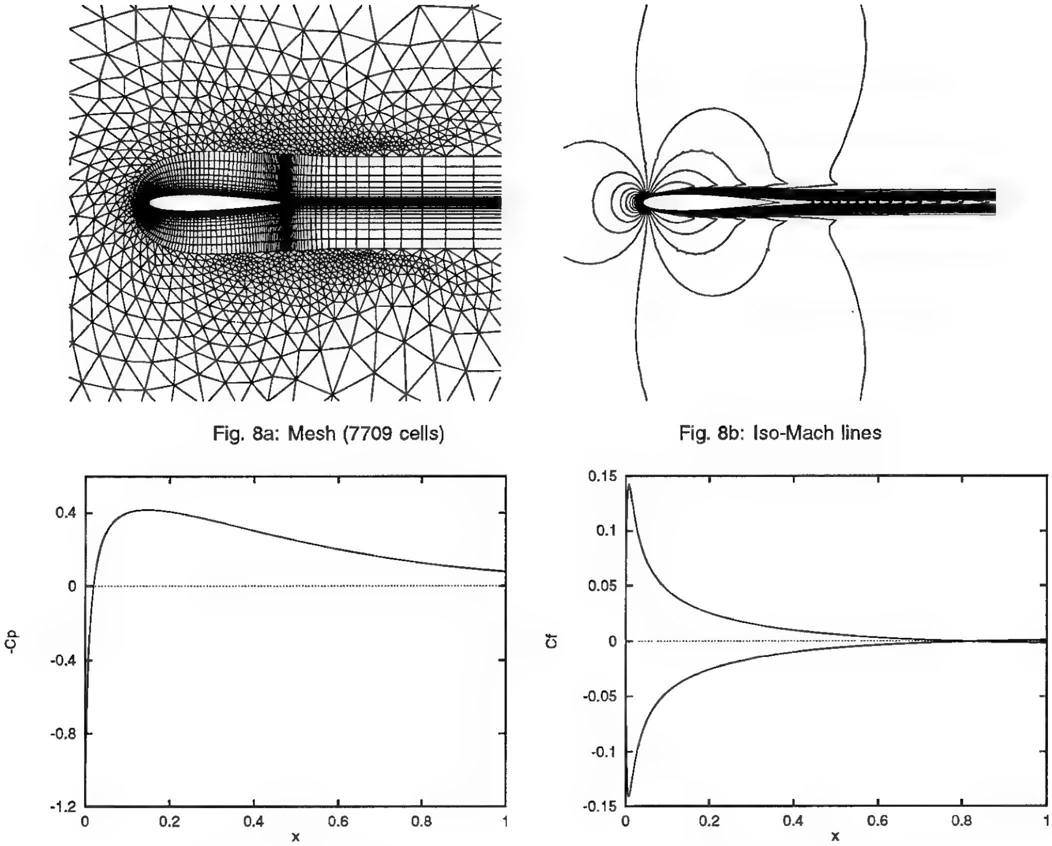


Fig. 8a: Mesh (7709 cells)

Fig. 8b: Iso-Mach lines

Fig. 8c: Pressure coeff.

Fig. 8d: Skin friction coeff.

NACA0012 airfoil $M_\infty = 0.5$, $Re_\infty = 5000$, incidence = 0 deg

Un Schéma Cinétique d'Ordre 2 Préservant les Positivités pour les Equations d'Euler Compressibles sur Maillages non Structurés Auto-Adaptatifs

Ph. Villedieu, J.L. Estivalezes, J.J. Hylkema

CERT-ONERA , 2 Ave. Ed. Belin, 31055 Toulouse, FRANCE

Abstract

The aim of this contribution is to present the first numerical results, that we have obtained with a new second order kinetic theory based scheme. The main interest of our approach is that density and internal energy can be proved to remain non negative under a CFL like condition. It is well known that classical approximate Riemann solvers, even first order accurate, do not satisfy this property. Our first order scheme is the classical kinetic scheme, based on the Maxwellian velocity distribution. Our second order extension consists of adding to the first order numerical flux an antidiffusive correction which has to be limited such that the constraints of positivity will be satisfied. It can be seen as a variant of the so-called corrected anti-diffusive flux approach. We have performed numerical computations for various two and three dimensionnal test cases on unstructured and self-adaptative meshes, in order to evaluate the accuracy and the robustness of this new method. Comparisons have been done with a second order extension of Roe's scheme (with MUSCL approach).

Introduction

Le but de cette contribution est de présenter les premiers résultats numériques obtenus avec un nouveau schéma d'ordre 2, basé sur la théorie cinétique des gaz. Le principal intéret de notre approche réside dans le fait qu'on peut prouver que la densité et la pression restent positives sous une condtion de type CFL. Il est bien connu que les schémas classiques, construits sur la base d'un solveur de Riemann approché, ne possèdent pas cette propriété même à l'ordre 1 [3]. C'est un sérieux inconvénient lorsqu'on souhaite calculer des écoulements pour lesquels la densite est tres faible ou pour lesquels l'energie interne est faible devant l'energie cinétique (Ecoulements hypersoniques, Problèmes de détonique, ...). Noter de plus que notre approche peut se généraliser sans difficulté aux écoulements réactifs [12,16].

Au premier ordre, notre schéma n'est autre que le schéma cinétique classique basé sur la distribution de vitesses Maxwellienne introduit par Pullin dans [19]. On peut montrer que ce schéma préserve la positivité de la densité et de la pression sous une condition de type CFL [11]. Notre extension au second ordre consiste à ajouter au flux numérique du premier ordre une correction anti-diffusive qui doit être limitée de sorte que les positivités soient préservées. Cette approche peut être considérée comme une variante de la méthode dite 'des flux modifiés'.

Les maillages utilisés sont structurés ou non structurés; de plus une technique de raffinement automatique de maillage a été implantée dans les codes 2D et 3D. Nous avons réalisé de nombreuses simulations numériques sur différents types de maillages, afin d'évaluer la robustesse et la précision de ce nouveau schéma. Des comparaisons ont également été faites avec le schéma de Roe étendu au second ordre suivant la méthode MUSCL de Van Leer.

Le plan de l'article est le suivant. On commence tout d'abord par quelques généralités sur les schémas cinétiques dont on rappelle les principales propriétés. Dans la seconde partie, on présente le principe de notre extension au second ordre. La troisième partie est consacrée a l'exposé d'un critère de raffinement de maillage (basé sur la production locale d'entropie du schéma) et à la description de la technique de raffinement de maillages que nous avons utilisée. Enfin dans la dernière partie, on présente de nombreux résultats numériques et des comparaisons avec le schéma de Roe.

1 Généralités sur les schémas cinétiques

Le premier schéma cinétique pour les équations d'Euler a été introduit par D. Pullin dans [19]. Il a ensuite été revisité et amélioré par S. Deshpande dans [4,5]. D'autres schémas cinétiques, basés sur des distributions d'équilibre différentes de la Maxwellienne, ont ensuite été proposés par divers auteurs dont Kaniel et Perthame [7,8,9]. Le grand intéret des travaux de B. Perthame est d'avoir les premiers mis en évidence les propriétés théoriques de certains schémas cinétiques (ceux associés à des distributions d'équilibre à support compact):consistance avec l'inéquation d'entropie, préservation des positivités, ... Signalons enfin les travaux de Mazet et al concernant les liens entre les schémas cinétiques et la symétrisation des équations d'Euler via les variables entropiques [1,12,11]. Nous reviendrons sur cet aspect dans la troisième partie de cet article.

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

1.1 Définition d'un schéma cinétique

Les schémas cinétiques sont des schémas Volumes-Finis décentrés pour lesquels la fonction Flux Numérique $\mathcal{F}(w,w',n)$ est de type Flux-Splitting c'est-à-dire de la forme suivante:

$$\mathcal{F}(w,w',n) = \mathcal{F}^{+}(w,n) + \mathcal{F}^{-}(w',n)$$

où w et w' sont deux états quelconques et n un vecteur unitaire de $\mathbf{R}^d$ (représentant la normale a l'interface entre deux cellules). Les fonctions $\mathcal{F}^{+}(w,n)$ et $\mathcal{F}^{-}(w,n)$ s'expriment sous la forme d'une intégrale sur l'espace dit des 'phases' en Physique Statistique. Dans le cas d'un gaz mono-atomique, on a par exemple:

$$\mathcal{F}^{\pm}(w,n) = \int_{\mathbf{R}^d} (\xi.n)^{\pm} \begin{pmatrix} 1 \\ \xi \\ 1/2|\xi|^2 \end{pmatrix} f_w(\xi) d\xi \qquad (1)$$

où $f_w(\xi)$ désigne la distribution d'équilibre des particules et satisfait par définition les relations:

$$w = \begin{pmatrix} \rho \\ \rho U \\ \frac{1}{2}\rho|U|^2 + \rho\epsilon(T) \end{pmatrix} = \int_{\mathbf{R}^d} \begin{pmatrix} 1 \\ \xi \\ \frac{1}{2}|\xi|^2 \end{pmatrix} f_w(\xi) d\xi$$

La formule (1) peut s'interpréter en considérant que les particules traversant une interface sont constituées de celles venant de gauche et se déplaçant dans le sens de la normale (contribution à F^{+}) et de celles venant de droite et se déplaçant en sens opposé à la normale (contribution à F^{-}).

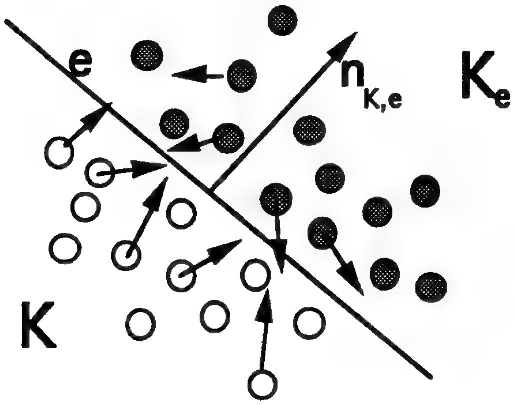


figure 1: Flux de particules à travers une interface

Différents choix sont possibles pour la fonction d'équilibre $f_w(\xi)$. Les deux plus courants sont la fonction 'créneau' proposée par Perthame [7,8]

$$f_w(\xi) = \frac{1}{Vol(B^d)} \frac{\rho}{(2rT)^{d/2}} Y(\frac{(\xi-u)^2}{2rT})$$

(où Y est la fonction indicatrice de $[0,1]$, B^d la boule unité de $\mathbf{R}^d$) et la fonction 'Maxwellienne' proposée par Pullin [19]

$$f_w(\xi) = \frac{\rho}{(2\pi rT)^{d/2}} \exp(\frac{(\xi-u)^2}{2rT})$$

C'est cette dernière que nous avons choisie car c'est la mieux adaptée pour l'extension aux mélanges de gaz réels [12] et c'est celle qui conduit aux formules les plus simples pour l'extension à l'ordre 2 [14].

On peut bien sûr expliciter la formule (1) dans le cas où f_w est une Maxwellienne. Dans le cas d'un gaz parfait d'équation $p = \rho rT$ avec $\epsilon = f(T)$ (f fonction régulière quelconque) on a:

$$\begin{aligned} \mathcal{F}^{\pm}(w,n) = {} & \pm h(X) \begin{pmatrix} \rho \\ \rho U \\ \frac{1}{2}\rho|U|^2 + \rho\epsilon(T) + \frac{1}{2}p \end{pmatrix} c \\ & + \tfrac{1}{2}(1 \pm g(X)) \begin{pmatrix} \rho U.n \\ \rho U U.n + pn \\ (\frac{1}{2}\rho|U|^2 + \rho\epsilon(T) + p)U.n \end{pmatrix} \end{aligned} \qquad (2)$$

où $c = \sqrt{2rT}$ $X = \frac{U.n}{c}$
$g(X) = \frac{2}{\sqrt{\pi}} \int_0^X \exp(-u^2) du$ $h(X) \doteq \frac{1}{2\sqrt{\pi}} \exp(-X^2)$

Pour une présentation plus complète des schémas cinétiques et en particulier pour la définition précise d'une distribution d'équilibre, on pourra se reporter par exemple à l'article de B. Perthame [7].

1.2 Quelques propriétés des schémas cinétiques

L'expression (1) des fonctions $\mathcal{F}^{+}$ et $\mathcal{F}^{-}$ permet de démontrer de nombreuses propriétés des schémas cinétiques. En particulier, en dimension 1 d'espace, lorsque le support de la distribution d'équilibre est un compact de la forme $[-\xi_{max}, \xi_{max}]$, on peut montrer [7] que les schémas cinétiques associés sont entropiques et préservent la positivité de la densité et de la température sous la condition CFL $\Delta t < \Delta x/\xi_{max}$.

Dans le cas d'une distribution Maxwellienne (support non borné), le problème est plus délicat et, à notre connaissance, la consistance avec l'inéquation d'entropie a seulement été prouvée de manière formelle dans [5,11]. La question de la positivité du schéma a par contre été résolue dans [11,14]. On rappelle ci-dessous le principal résultat.

Pour être le plus général possible, on se place dans un cadre multidimensionnel (d désignant le nombre de dimensions d'espace) et le maillage, noté $\mathcal{M}_h$, est supposé quelconque. On note K un élément quelconque de $\mathcal{M}_h$, m(K) sa mesure de Lebesgue dans $\mathbf{R}^d$, K_e son voisin le long de la face e, $n_{K,e}$ la normale à la face e dirigée de K vers K_e et $m(e)$ la mesure de Lebesgue de la face e dans $\mathbf{R}^{d-1}$ (cf figure 2).

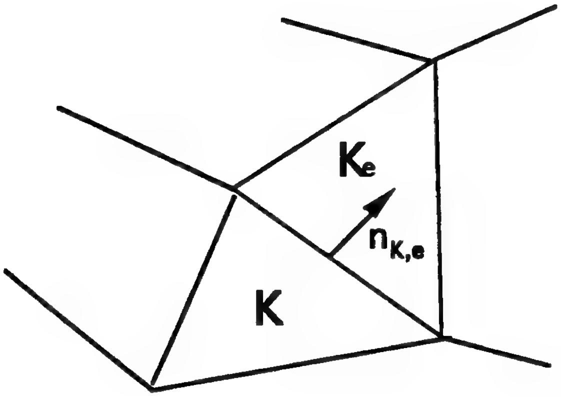


figure 2: Vue partielle du maillage

On montre dans [14] la proposition suivante:

Proposition 1 *Le schéma Volumes-Finis*

$$w_K^{n+1} = w_K^n - \frac{\Delta t}{m(K)} \sum_{e\in\partial K} \left[\mathcal{F}^+(w_K^n, n_{K,e}) + \mathcal{F}^-(w_{K_e}^n, n_{K,e})\right] m(e)$$

associé au flux numérique défini par les formules (2) (distribution Maxwellienne) preserve la positivité de ρ et de T sous la condition CFL

$$\sup_{K\in\mathcal{M}_h} \frac{m(\partial K)(|U_K^n| + \frac{2d+1}{d}\sqrt{\frac{rT_K^n}{2\pi}})}{m(K)}\Delta t < 1 \qquad (3)$$

Remarque: En pratique cette condition est un peu plus restrictive que la condition CFL usuelle (elle correspond environ pour un gaz parfait avec $\gamma = 1.4$ à $CFL = 0.5$). Toutefois c'est seulement une condition suffisante et, dans les applications, on n'a jamais constaté de difficulté en prenant $CFL = 0.9$.

2 Extension a l'ordre 2 et positivité

2.1 Principe de l'extension à l'ordre 2

Pour étendre une méthode de volumes finis à l'ordre 2 en espace, il existe au moins deux approches classiques:

- La première (sans doute la plus utilisée du fait de sa simplicité et de sa généralité) est la méthode MUSCL de Van Leer. Elle consiste à décomposer un pas de temps en deux étapes: une première étape d'interpolation affine de la solution approchée, une seconde étape où l'on applique le schéma volume fini aux valeurs interpolées de la solution approchée. Le point essentiel réside dans le fait que lors de l'étape d'interpolation, il est nécessaire de limiter la valeur du gradient de la solution approchée afin d'éviter l'apparition d'oscillations.
- La seconde désignée dans la littérature anglo-saxone sous le non de 'corrected antidiffusive flux approach' (elle sera notée CAFA par la suite) consiste à ajouter au flux numérique du premier ordre une correction antidiffusive qui doit être limitée pour des raisons de stabilité numérique.

Ces deux méthodes ont été très bien étudiées d'un point théorique dans le cas dans le cas d'une loi de conservation scalaire (voir par exemple Goldveski-Raviart [6], Coquel-Lefloch [2], ...). En particulier, on sait dans ce cas donner des critères précis sur la façon dont les pentes doivent être limitées pour que la méthode numérique soit stable au sens de la norme BV (schéma TVB) ou de la norme L^∞.

Dans le cas des systèmes généraux de lois de conservation, en particulier celui des équations d'Euler, il n'existe actuellement aucune théorie. On se contente donc en général de raisonner par analogie avec le cas scalaire, afin d'en déduire certains critères empiriques de stabilité. De plus, dans le cas de la dynamique des gaz, vient se rajouter le fait que la solution $w = (\rho, \rho U, \rho E)$ ne prend pas ses valeurs dans R^{d+2} tout entier mais seulement dans un sous ensemble de celui-ci, W_{ad}, défini par des contraintes de positivité $\rho > 0$, $\rho E - 1/2\rho U^2 > 0$. Quelque soit l'approche adoptée, MUSCL ou CAFA, il est nécessaire de joindre (ou éventuellement de substituer) à ces critères empiriques de limitation de pente ou de flux, qui permettent de contrôler les oscillations, une condition qui garantisse que le schéma laisse invariant l'ensemble W_{ad} (ce qui suppose bien sûr que la propriété est dejà satisfaite par le schéma d'ordre 1). Il est intéressant de remarquer que cette seule propriété d'invariance de l'ensemble W_{ad} garantit la stabilité en norme L^1 du schéma [9] et constitue donc un critère de stabilité faible.

Dans le cas de la méthode MUSCL, une variante a été proposée par Perthame et Qiu qui garantit la préservation des positivités. Le principe est de construire la solution interpolée à chaque pas de temps de sorte que l'on ait à la fois conservation de $\rho, \rho U, \rho E$ sur chaque cellule et positivité de ρ et T aux noeuds du maillage [10]. Cependant les résultats numériques sont assez décevant du point de vue du gain en précision, et leur technique de reconstruction semble difficilement généralisable sur des maillages quelconques.

L'approche que nous proposons dans cet article est plutôt une variante de la méthode 'CAFA'. Elle peut a-priori s'étendre à tout schéma de Flux-Splitting (cet aspect est dévellopé dans [14]) mais les schémas cinétiques possèdent toutefois deux avantages essentiels:

- La positivité peut être prouvée dans le cas du schéma d'ordre 1 ce qui n'est pas le cas par exemple pour les schémas de Flux-Splitting de Steger et Warming ou de Van Leer.
- Il est possible, grâce à la représentation intégrale (1) du flux numérique, d'expliciter les limitations à imposer sur les corrections antidiffusives pour que le schéma préserve les positivités.

Nous allons rappeler ici les grandes lignes de cette approche, en renvoyant à [14] pour les détails. L'idée générale est de remplacer sur chacune des interfaces e du maillage $\mathcal{F}^+(w_K, n_{K,e})$ par $\mathcal{F}^+(w_K, n_{K,e}) + \Delta\mathcal{F}^{n+}_{K,e}$ et $\mathcal{F}^-(w_{K_e}, n_{K,e})$ par $\mathcal{F}^-(w_{K_e}, n_{K,e}) + \Delta\mathcal{F}^{n-}_{K,e}$, où les fonctions $\mathcal{F}^+$ et $\mathcal{F}^-$ sont définies par les formules (2) (correspondant à une distribution Maxwellienne) et $\Delta\mathcal{F}^+$ et $\Delta\mathcal{F}^-$ sont des corrections antidiffusives qui pour un gaz mono-atomique s'expriment sous la forme intégrale suivante:

$$\Delta\mathcal{F}^{n+}_{K,e} = \int_{\mathbf{R}^d} (\xi.n_{K,e})^+ \begin{pmatrix} 1 \\ \xi \\ 1/2|\xi|^2 \end{pmatrix} \Delta f_w(\xi)^n_{K,e} d\xi \quad (4)$$

$$\Delta\mathcal{F}^{n-}_{K,e} = \int_{\mathbf{R}^d} (\xi.n_{K,e})^- \begin{pmatrix} 1 \\ \xi \\ 1/2|\xi|^2 \end{pmatrix} \Delta f_w(\xi)^n_{K_e,e} d\xi \quad (5)$$

où $\Delta f_w(\xi)$ est une correction de la distribution d'équilibre, fonction comme dans le développement de Chapmann et Enskog [18] des gradients locaux de la solution. Dans le cas où f_w est une Maxwellienne, elle s'exprime sous la forme suivante (cf [14]):

$$\begin{aligned}\Delta f_w(\xi)^n_{K,e} &= \Bigg((\delta s)^n_{K,e} + \sum_{j=1}^{d} Y(\overline{\xi^j})\overline{\xi^j}(\delta U^j)^n_{K,e} \\ &+ 1/2(\delta T)^n_{K,e} \sum_{j=1}^{d} Y(\overline{\xi^j})\overline{\xi^j}^2 \Bigg) f_{w^n_K}(\xi)\end{aligned}$$

$$\overline{\xi^j} = \frac{\xi^j - U^j}{\sqrt{2rT^n_K}}$$

$$Y(x) = 1 \ \ si \ \ |x| \le \overline{\xi}_{max}, \ \ 0 \ \ si \ \ |x| > \overline{\xi}_{max}$$

$$(\delta s)^n_{K,e} = \frac{-1}{r}(ps)^n_K.(x_e - x_K) - \frac{1}{r}(qs)^n_K \frac{k^n}{2}$$

$$(\delta U^j)^n_{K,e} = \sqrt{\frac{2}{rT}}(pU^j)^n_K.(x_e - x_K) + \sqrt{\frac{2}{rT}}(qU^j)^n_K \frac{k^n}{2}$$

$$(\delta T)^n_{K,e} = \frac{1}{T}(pT)^n_K.(x_e - x_K) + \frac{1}{T}(qT)^n_K \frac{k^n}{2}$$

où x_e désigne le centre de la face e, x_K le centre de gravité de la cellule K, $(ps)^n_K$, $(pU^j)^n_K$, et $(pT)^n_K$ des estimations des gradients spatiaux respectifs de s,U^j et T dans la cellule K à l'instant n et $(qs)^n_K$, $(qU^j)^n_K$, et $(qT)^n_K$ des estimations des dérivées temporelles de s,U^j et T dans la cellule K à l'instant n. Nous préciserons au paragraphe suivant les choix effectués pour calculer ces quantités ainsi que la valeur de $\overline{\xi}_{max}$

NB: Noter que δs, δU et δT sont des quantités en $O(h)$ (h étant le pas du maillage) lorsque la solution est régulière.

Du point de vue cinétique, on peut interpréter les formules (4) et (5) en considérant que la distribution d'équilibre Maxwellienne $f_{w^n_K}(\xi)$ a été remplacée par une distribution d'équilibre 'perturbée' $f_{w^n_K}(\xi) + \Delta f_{w^n_K}(\xi)$, tenant compte des gradients de la solution (ce point de vue est exposé dans [15]). Cette idée était déjà présente dans les travaux de S. Deshpande [5]. L'introduction d'une fonction créneau $Y(x)$ dans la définition de Δf_w permet de garantir que la distribution modifiée $f_{w^n_K}(\xi) + \Delta f_{w^n_K}(\xi)$ reste toujours positive (à condition de choisir convenablement $\overline{\xi}_{max}$). Cette propriété joue un rôle essentiel pour la positivité du schéma d'ordre 2.

On montre dans [14] le résultat suivant:

Proposition 2 *Si pour tout $K \in \mathcal{M}_h$ et pour tout $e \in K$:*

(i) $|(\delta s)^n_{K,e}| \leq 1$

(ii) $\overline{\xi}_{max} =$

$$\frac{-|(\delta U)^n_{K,e}| + \sqrt{{(\delta U)^n_{K,e}}^2 + 4|(\delta T)^n_{K,e}|(1-|(\delta s)^n_{K,e}|)}}{2\sqrt{d}|(\delta T)^n_{K,e}|}$$

avec $|(\delta U)^n_{K,e}| = \sqrt{\sum_{j=1}^{d} |(\delta U^j)^n_{K,e}|^2}$, *le schéma décrit ci-dessus est du second ordre en temps et en espace si* $\mathcal{M}_h$ *est une grille cartésienne et si les gradients sont estimées au second ordre. De plus, il préserve la positivité de* ρ *et de* T *sous la condition CFL:*

$$\sup_{K \in \mathcal{M}_h} \frac{m(\partial K)2(|U^n_K| + \frac{2d+1}{d}\sqrt{\frac{rT^n_K}{2\pi}})}{m(K)}\Delta t < 1$$

Remarques:

- On remarque qu'aucune limitation de type 'min-mod' n'est nécessaire pour garantir la positivité du schéma. On en donne une illustration numérique dans [14]. Toutefois les limitations (i) et (ii) ne sont pas suffisantes, pour contrôler totalement l'apparition d'oscillations spatiales. Il est nécessaire en pratique de les associer à d'autres limitations plus classique de type 'min-mod' qui seront explicitées au paragraphe suivant.

- La condition CFL ci-dessus est un peu trop restrictive. Dans les applications, nous n'avons jamais constaté de difficulté en prenant $CFL = 0.9$.

- Tous les résultats exposés ci-dessus se généralisent dans le cas d'un gaz parfait polyatomique de γ quelconque. Il suffit d'augmenter la dimension de l'espace des phases pour prendre en compte les degrés de liberté internes des molécules. La condition de positivité fait cette fois intervenir une contrainte supplémentaire sur $|\delta T|$. On renvoie à [14,13] pour les détails et les formules explicites permettant de calculer $\Delta\mathcal{F}^+$ et $\Delta\mathcal{F}^-$.

2.2 Principe du calcul et de la limitation des gradients de la solution discrète

De nombreuses solutions sont proposées dans la littérature pour estimer les gradients de la solution discrète à partir de ses valeurs dans chacune des cellules du maillage. Par souci de simplicité et également pour des raisons liées à notre structure de données, nous avons choisi la formule suivante:

$$\nabla q_K = \frac{1}{2m(K)} \sum_{e \in \partial K} (q_K + q_{K_e}) n_{K,e} m(e) \qquad (6)$$

où q désigne une composante quelconque de w. Cette formule est d'ordre 2 en h si la solution est régulière et si le maillage est cartésien.

Pour limiter les gradients obtenus nous avons utilisé une généralisation multidimensionnelle du limiteur 'min-mod', qui consiste à imposer que les valeurs Min et Max de la fonction $q(x) = q_k + \nabla q_K.(x - x_K)$ au centre des faces de l'élément K (et non pas aux sommets de l'élément K, ce qui serait plus contraignant) soient comprises entre le Min et le Max des valeurs de q sur les éléments voisins de K. En pratique on commence par calculer les valeurs Min et Max de $q(x)$ sur l'élément K (notées q_{min} et q_{max}) ainsi que les valeurs Min et Max de q sur les éléments voisins (notées $\tilde{q}_{min}$ et $\tilde{q}_{max}$). On pose ensuite:

$$\alpha_{max} = \text{Max}\left(0, \frac{\tilde{q}_{max} - q_K}{q_{max} - q_K}\right)$$
$$\alpha_{min} = \text{Max}\left(0, \frac{\tilde{q}_{min} - q_K}{q_{min} - q_K}\right)$$
$$\alpha = \text{Min}(1, \alpha_{min}, \alpha_{max})$$

On prend enfin:

$$\nabla q^{lim}_K = \alpha \nabla q_K$$

L'estimation des dérivées en temps de la solution discrète se fait à partir de la forme non conservative des équations d'Euler. Sous forme non conservative, le système de la dynamique des gaz peut en effet s'écrire:

$$\partial_t s = -U.\nabla s$$

$$\partial_t U = -U \otimes \nabla U - \frac{1}{\rho}\nabla p$$

$$\partial_t T = -U.\nabla T - (\gamma - 1)T\text{div}U$$

On obtient les estimations souhaitées de $\partial_t s$, $\partial_t U$ et $\partial_t T$ en se servant de ces relations et des valeurs de ∇s, ∇p, ∇T et ∇U, calculées selon la formule (6).

3 Critère de raffinement et maillages auto-adaptatifs

3.1 Description de la technique de raffinement de maillages

Afin d'améliorer la précision des résultats pour le calcul d'écoulements stationnaires, une procédure de raffinement automatique de maillages a été implantée dans les codes de calcul 2D et 3D. Le principe utilisé est le suivant:

1. On commence par calculer une première solution stationnaire sur le maillage grossier de départ.
2. On calcule alors sur chacun des éléments du maillage la valeur du critère de raffinement et en fonction de ce critère certains éléments sont raffinés selon une procédure qui sera décrite ci-dessous.
3. On calcule ensuite une nouvelle solution sur le maillage raffiné en partant de la solution sur le maillage précédent.
4. On réitère éventuellement le processus

On utilise, dans notre code de calcul 2 types d'éléments en dimension 2 (triangles et quadrangles) et 3 en dimension 3 (tétraèdres, pentaèdres et hexaèdres). La répartition de ces différents éléments peut être quelconque au sein d'un même maillage. En particulier il n'est pas nécessaire d'assurer la coincidence nodale entre deux éléments voisins comme l'illustre par exemple la figure 3.

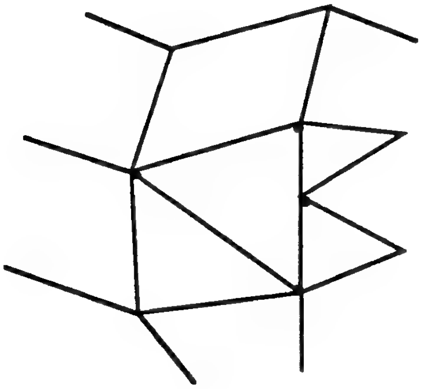


figure 3: Vue partielle d'un maillage non conforme

Pour raffiner un élément, le principe consiste dans tous les cas à le diviser en un certain nombre d'éléments fils (4 en dimension 2, 8 en dimension 3) tous semblables à l'élément de départ. Chaque élément est bien sûr raffiné indépendamment de ses voisins, si bien qu'à l'issue d'une phase de raffinement le maillage obtenu est généralement non conforme (non coincidence nodale entre certains éléments). On a schématisé sur les figures 4 et 5 le principe de raffinement d'un triangle et d'un quadrangle. En dimension 3, on renvoie pour une description détaillée de la procédure de raffinement au travail de J. Delaire [20].

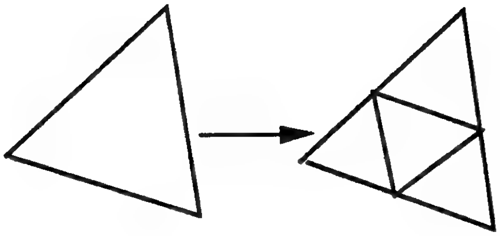


figure 4: Raffinement d'un triangle

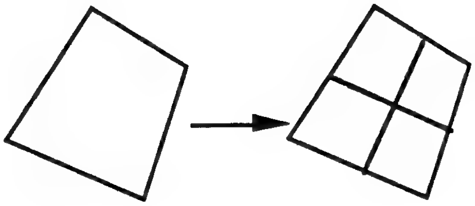


figure 5: Raffinement d'un quadrangle

3.2 Un critère de raffinement fondé sur la production locale d'entropie

Nous allons maintenant décrire le critère de raffinement utilisé. Ce critère a été introduit par P. Mazet et al dans [1,11]. Il repose sur les liens entre les schémas cinétiques et la symétrisation, via les variables entropiques, des équations d'Euler.

Commençons par quelques rappels sur la symétrisation des équations d'Euler. Afin de simplifier les notations, on se restreint au cas d'un gaz parfait polytropique. Soit $s = r\log(\frac{T^{1/(\gamma-1)}}{\tau})$ l'entropie massique du gaz. Il est bien connu que la fonction $S(w) = -\rho s$ (où w est le vecteur des variables conservatives) est une fonction strictement convexe en w et constitue une entropie de Lax pour les équations d'Euler, associée au flux d'entropie $US(w)$. Les variables entropiques ϕ_α sont alors définies de la manière suivante:

$$\begin{aligned}\phi_\rho &= \frac{\partial S}{\partial \rho} = c_p - s - \frac{U^2}{2T} \\ \phi_{\rho U^j} &= \frac{\partial S}{\partial \rho U^j} = \frac{U^j}{T} \quad (j = 1, \dots, d) \\ \phi_{\rho E} &= \frac{\partial S}{\partial \rho E} = -\frac{1}{T}\end{aligned} \tag{7}$$

Les relations (7) définissent un changement de variables bijectif $w \rightarrow \phi$ de W_{ad} sur $\Phi_{ad} = \mathbf{R}^{d+1} \times \mathbf{R}^{+\star}$. On définit alors sur Φ_{ad} la transformée de Legendre $S^\star$ de S par:

$$S^\star(\phi) = -S(w(\phi)) + w(\phi).\phi$$

De même, on introduit une pseudo-transformée de Legendre du flux d'entropie dans la direction du vecteur n, notée $\Sigma^\star(\phi, n)$, en posant:

$$\Sigma^\star(\phi, n) = -U.nS(w(\phi)) + (F(w(\phi)).n).\phi$$

où $F(w).n = [\rho U.n, \rho UU.n + pn, (\rho E + p)U.n]^t$ désigne le flux dans la direction n. Cette fonction est appelée fonction de symétrisation du système des équations d'Euler car elle possède par construction la propriété suivante:

$$F(w(\phi)).n = \nabla_\phi \Sigma^\star(\phi, n) \tag{8}$$

Le flux des équations d'Euler, exprimé en variables entropiques, est donc le gradient de la fonction de symétrisation $\Sigma^\star(\phi, n)$. Il en découle immédiatement, qu'écrit en variables entropiques, le système de la dynamique des gaz est symétrique. D'autre part, d'après (8) toute décomposition de la fonction $\Sigma^\star(\phi, n)$ en la somme de deux fonctions $\Sigma^{\star+}(\phi, n)$ et $\Sigma^{\star-}(\phi, n)$ induit, en différenciant par rapport à ϕ, une décomposition du flux $F(w).n$ en la somme de deux fonctions $F^+(w, n)$ et $F^-(w, n)$. Si de plus, pour tout n, $\Sigma^{\star+}(\phi, n)$ est convexe en ϕ et $\Sigma^{\star-}(\phi, n)$ concave, alors on peut montrer (voir [11]) que la jacobienne de $F^+(w, n)$ est diagonalisable à valeurs propres positives et que la jacobienne de $F^-(w, n)$ est diagonalisable à valeurs propres négatives. La décomposition en partie convexe et concave d'une fonction de symétrisation fournit donc un moyen naturel de construire des schémas de Flux-Splitting correctement décentré.

Le lien avec le formalisme cinétique provient du fait que la fonction $\Sigma^\star$ peut s'écrire, à une constante multiplicative près, sous la forme intégrale suivante (cas d'un gaz mono-atomique):

$$\Sigma^\star(\phi, n) = \int_{\mathbf{R}^d} (\xi.n) \exp[(\phi_\rho + \phi_{\rho U}.\xi + \phi_{\rho E}.\frac{\xi^2}{2})/r] d\xi \tag{9}$$

La fonction $\exp[(\phi_\rho + \phi_{\rho U}.\xi + \phi_{\rho E}.\frac{\xi^2}{2})/r]$ n'est autre que la Maxwellienne. La fonction exp étant convexe sur $\mathbf{R}$, on obtient une décomposition convexe-concave de la fonction $\Sigma^\star$ en posant simplement:

$$\Sigma^{\star\pm}(\phi, n) = \int_{\mathbf{R}^d} (\xi.n)^\pm \exp[(\phi_\rho + \phi_{\rho U}.\xi + \phi_{\rho E}.\frac{\xi^2}{2})/r] d\xi \tag{10}$$

En différenciant (10) par rapport à ϕ on voit facilement que les fonctions $F^+(w, n)$ et $F^-(w, n)$ ainsi obtenues ne sont autres que les fonctions $\mathcal{F}^+(w, n)$ et $\mathcal{F}^-(w, n)$ définies par (1). On peut donc écrire que:

$$\mathcal{F}^\pm(w, n) = \nabla_\phi \Sigma^{\star\pm}(\phi(w), n) \tag{11}$$

Nous allons maintenant utilisé la propriété (11) et la concavité de la fonction $\Sigma^{\star-}(\phi(w), n)$ pour établir une estimation de la production locale d'entropie du schéma cinétique d'ordre 1, introduit à la section 1, lorsque l'état stationnaire est atteint. Par définition, lorsque l'état stationnaire est atteint , on a sur toutes les cellules K du maillage:

$$\sum_{e \in \partial K} [\mathcal{F}^+(w_K, n_{K,e}) + \mathcal{F}^-(w_{K_e}, n_{K,e})]\, m(e) = 0$$

En se servant du fait que $\sum_{e \in \partial K} F(w_K).n_{K,e} m(e) = 0$ (formule de Green), et en ajoutant cette quantité au premier membre de la précédente égalité on obtient:

$$\sum_{e \in \partial K} [\mathcal{F}^-(w_{K_e}, n_{K,e}) - \mathcal{F}^-(w_K, n_{K,e})]\, m(e) = 0 \tag{12}$$

D'autre part, posons:

$$\Sigma^\pm(w, n) = -\Sigma^{\star\pm}(\phi(w), n) + \mathcal{F}^\pm(w).\phi(w) \tag{13}$$

On déduit de la définition de $\Sigma^\star$ que:

$$\Sigma^+(w, n) + \Sigma^-(w, n) = U.nS(w)$$

Le couple $[\Sigma^+(w, n), \Sigma^-(w, n)]$ constitue donc une décomposition du flux d'entropie. En multipliant scalairement l'égalité (12) par $\phi_K = \phi(w_K)$ et en se servant de (13), (11) et de la formule de Green, on obtient:

$$\sum_{e \in \partial K} [\Sigma^+(w_K, n_{K,e}) + \Sigma^-(w_{K_e}, n_{K,e})]\, m(e) - Q_K = 0 \tag{14}$$

avec:

$$\begin{aligned} Q_K = \sum_{e \in \partial K} m(e) \, [&\Sigma^{\star-}(\phi_K, n_{k,e}) - \Sigma^{\star-}(\phi_{K_e}, n_{k,e}) \\ &-\nabla_\phi \Sigma^{\star-}(\phi_{K_e}, n_{k,e}) (\phi_K - \phi_{K_e})] \end{aligned}$$

La fonction $\Sigma^{\star -}(.,n)$ étant concave, le terme Q_K est négatif. Il peut s'interpréter comme la production locale d'entropie sur la cellule K, l'autre terme dans (14) étant simplement un terme de flux. C'est cette quantité que nous avons utilisée comme critère de raffinement, le calcul explicite de la fonction $\Sigma^{\star -}$ pouvant s'effectué facilement grâce à la définition (10).

4 Schéma implicite

Pour le calcul d'écoulement stationnaire 3D, l'utilisation d'un schéma explicite en temps s'est avérée trop coûteuse, même avec une technique de pas de temps local. Le schéma décrit à la section 2 a donc été implicité, suivant le principe classique qui consiste à linéarisé partiellement le système non linéaire que l'on doit résoudre à chaque pas de temps. De ce point de vue, les schémas cinétiques possèdent une particularité intéressante:les fonctions $\mathcal{F}^-$ et $\mathcal{F}^+$ sont différentiables et homogènes de degré 1. Elle vérifient donc les relations suivantes:

$$\begin{array}{l} \mathcal{F}^+(w,n) = [Jac(\mathcal{F}^+)(w,n)].w \\ \mathcal{F}^-(w,n) = [Jac(\mathcal{F}^-)(w,n)].w \end{array} \qquad (15)$$

ce qui permet de simplifier l'écriture de la forme linéarisée du schéma implicite. Celle-ci peut donc s'écrire:

$$\begin{array}{l} \left[Id + \frac{\Delta t}{m(K)} \sum_{e \in \partial K} [Jac(\mathcal{F}^+)(w_K^n, n_{K,e})] m(e) \right] .w_K^{n+1} \\ + \frac{\Delta t}{m(K)} \sum_{e \in \partial K} [Jac(\mathcal{F}^-)(w_{K_e}^n, n_{K,e})].w_{K_e}^{n+1} m(e) \\ = w_K^n - \frac{\Delta t}{m(K)} \sum_{e \in \partial K} (\Delta \mathcal{F}_K^{n+} + \Delta \mathcal{F}_{K,e}^{n-}) m(e) \end{array}$$

La correction du second ordre n'est pas implicitée, afin de simplifier l'expression de la matrice Jacobienne du flux numérique. On n'a pas rencontré pour autant de problème de stabilité.

A chaque pas de temps, le système linéaire ci-dessus est résolu par une méthode itérative. On en a comparé deux: la méthode de Jacobi et la méthode BICGstab [17]. Si on se contente d'une précision moyenne à chaque résolution (ce qui est suffisant en pratique), la méthode de Jacobi est un peu plus performante en temps CPU. La tendance s'inverse si on souhaite une très grande précision. L'utilisation d'un préconditionneur pour la méthode BICGstab améliore nettement la vitesse de convergence (d'un facteur 2 au moins) mais n'apporte pas un gain en temps CPU compte tenu du coût du préconditionnement.

5 Résultats numériques

Afin d'évaluer la précision et la robustesse de ce nouveau schéma, nous avons réalisé plusieurs expériences numériques.

Tout d'abord, afin d'illustrer numériquement la préservation des positivités, nous avons calculé la solution du problème de Riemann proposé par Sjogreen dans [3], pour lequel la solution est très proche du vide. Nous avons effectué les calculs sans utiliser de limiteur de pentes de type min-mod (cf figure 6) puis avec limiteur (cf figure 7). Les résultats obtenus sans limitation sont bons mais font apparaître quelques oscillations, qui disparaissent avec l'utilisation du limiteur. Pour des cas test plus complexe (présence de discontinuités) l'utilisation d'un limiteur est nécessaire.

Sur les figures 8 et 9, on présente les résultats concernant le cas d'un écoulement hypersonique 2D à Mach 25 et Incidence 30. Le maillage a été obtenu après trois raffinements successifs. La méthode de Roe n'a pu être utilisée qu'à l'ordre 1 car à l'ordre 2, même avec de fortes limitations de pentes, des températures négatives apparaissent à l'arrière corps. Les résultats obtenus avec le schéma cinétique d'ordre 2 sont très bons et bien meilleurs que ceux obtenus avec le schéma de Roe. En particulier, on peut constater l'absence d'oscillations sur les courbes du coefficient de pression à la paroi (figure 8), contrairement aux résultats obtenus avec le schéma de Roe premier ordre (avec correction d'entropie).

Sur les figures 10 à 12, on présente les résultats concernant le calcul d'un écoulement instationnaire entrant à Mach 3 dans un tunel comportant une marche. Ce cas test est très classique et bien documenté. On compare les résultats obtenus avec la méthode de Roe-MUSCL (avec correction entropique) et le schéma cinétique. (sans correction entropique). Les deux maillages utilisés sont composés de triangles (le pas choisi est de $1/40^{eme}$ ce qui est assez grossier pour ce cas). Le second maillage a été reaffiné près de la paroi afin d'éliminer l'influence de la couche limite numérique. On peut noter que, sur les deux maillage, la détente sonique est mieux capturée avec le schéma cinétique qu'avec le schéma de Roe. On constate de plus la présences de nombreuses oscillations sur les courbes iso-densité obtenus avec le schéma de Roe. (Celles-ci disparaissent si on renforce les limitations de pentes). Par contre, avec le schéma cinétique, la position de la ligne de choc après la deuxième réflexion n'est pas correcte. Ce défaut semble du à la trop grande épaisseur de la couche limite numérique. Il disparaît lorsqu'on raffine le maillage près de la paroi. Avec le schéma de Roe, on constate que les oscillations numériques présentes au niveau de la ligne de glissement sont amplifiées lorsqu'on raffine le maillage au niveau de la paroi.

Sur les figures 13 et 14, on présente le calcul d'un écoulement stationnaire entrant à Mach 2 dans un tunel comportant une rampe inclinée à 15 degrés. Le maillage a été obtenu après trois raffinements successifs à partir d'un maillage grossier comportant 2500 cellules. Le critère de raffinement utilisé (cf section 4) a permis de detecter toutes les ondes présentes dans l'écoulement; en particulier le maillage a été raffiné au niveau de la ligne de glissement émanant du point triple situé sur la paroi supérieure (cf figure 11). On voit sur la figure 12 que le raffinement du maillage a permis une amélioration sensible de la qualité des résultats.

Enfin on présente sur les figures 15 à 17 des résultats numériques 3D concernant le calcul d'un écoulement transonique (Mach: 0.84, Incidence: 3.06 degrés) autour de la voilure M6 de l'ONERA. Ce cas test est très bien documenté dans [21]. Le maillage initial est constitué d'environ 60000 tétraèdres, ce qui est assez grossier pour ce type de calculs. Le maillage final (figure 20) a été obtenu après 2 raffinements successifs. Les résultats obtenus sont tout à fait en accord avec ceux des différents contributeurs du workshop AGARD [21]. Le raffinement du maillage permet là encore d'améliorer sensiblement la précision des résultats.

6 Conclusion

On a présenté dans cet article un nouveau schéma cinétique d'ordre 2 préservant la positivité de la masse volumique et de la température sous condition CFL. Les premier résultats numériques obtenus sur maillages non structurés sont très bons et confirment les propriétés théoriques de robustesse du schéma. De plus l'estimation d'entropie discrète associée au schéma d'ordre 1 permet de dégager, de manière naturelle, un critère de raffinement de maillage fondé sur la production d'entropie locale. Ce critère semble un excellent candidat pour la capture des discontinuités stationnaires.

La suite de cette étude va consister a étendre ce schéma au calcul d'écoulements réactifs, pour lesquels la robustesse de la méthode numérique est un critère essentiel. Ce travail est en cours.

References

[1] F. BOURDEL, J.P. CROISILLE, P. DELORME, P. MAZET, *Sur l'approximation par éléments finis des systèmes hyperboliques K-diagonalisables, Application aux équations d'Euler et aux mélanges de gaz*, La Recherche Aérospatiale, 1989-5, 15-34

[2] F. COQUEL, P.G. LEFLOCH, *Convergence of finite difference schemes for scalar conservation laws in several space dimensions: the corrected antidiffusive flux approach*, Math. of Comp. 57, 1991, p 169-210

[3] B. EINFELDT, C.D. MUNZ, P.L. ROE and B. SJOGREEN, *On godunov type methods near low densities*, J. of Comp. Phys., Vol. 92, 1991, 273-295

[4] S.M. DESHPANDE, *On the Maxwellian distribution, symmetric form and the entropy conservation for the compressible Euler equations*, Tech. Rep. 2583, NASA Langley, 1986

[5] S.M. DESHPANDE, *A second order accurate, kinetic theory based, method for inviscid compressible flows*, Tech. Rep. 2613, NASA Langley, 1986

[6] E. GODLEWSKI, P.A. RAVIART, *Hyperbolic systems of conservation laws*, SMAI, 1990

[7] B. PERTHAME, *Boltzmann type schemes for gas dynamics and the entropy property*, SIAM J. Num. Anal., Vol 27-6, 1990, 1405-1421

[8] B. PERTHAME, F. CORON, *Numerical passage from kinetic to fluid equations*, SIAM J. Num. Anal., Vol 28-1, 1991, 26-42

[9] B. PERTHAME, *Second order Boltzmann schemes for compressible Euler equations in one and two space variables*, SIAM J. Num. Anal., Vol. 29-1, 1992

[10] B. PERTHAME, Y. QIU, *A new variant of Van Leer's method for multidimensional systems of conservation laws*, Rapport technique 1562, INRIA, 1991

[11] P. VILLEDIEU, P.A. MAZET, *Schémas cinétiques pour les équations d'Euler hors équilibre thermochimique*, à paraître dans la Recherche Aérospatiale

[12] P. VILLEDIEU, *Approximations de type cinétique du système hyperbolique de la dynamique des gaz hors équilibre thermochimique*, Thèse de l'université Paul Sabatier, 1994

[13] P. VILLEDIEU, *Schémas cinétiques d'ordre 2 sur maillages non structurés*, Rapport Technique DERI n 2/3526.00, Fev. 1995

[14] P. VILLEDIEU, J.L. ESTIVALEZES, *High order positivity preserving schemes for the compressible Euler equations*, à paraître dans SIAM J. Num. Anal.

[15] P. VILLEDIEU, J.L. ESTIVALEZES, *A second order positivity preserving scheme for the compressible Euler equations*, Proceedings de la 14eme ICNMFD, Bangalore, 1994

[16] P. VILLEDIEU, *A positivity preserving second order accurate kinetic scheme for chemically reactive flows*, en préparation

[17] H. A. VAN DER VORST, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymetric linear systems*, SIAM J. Num. Anal., Vol 13-2, 1992

[18] C. CERCIGNANI, *The Boltzmann equation and its applications*, Scottish Academic Press 1975, Reed. A.M.S. 64 (Springer)

[19] *D.I. PULLIN, Direct simulations methods for compressible inviscid ideal gas-flows*, J. of Comp. Phys., Vol. 34, 1980, 53-66

[20] J. DELAIRE, *Algorithmes de raffinement de maillage non structuré 3D*, Rapport de Stage DERI, Sept. 1995

[21] AGARD ADVISORY REPORT, *Test cases for inviscid flow field methods*, May 1985

[22] G. MEHLMAN *Etude de quelques problèmes liés aux écoulements en déséquilibre chimique et thermique* , Thèse de doctorat de l'Ecole Polytechnique, 1991.

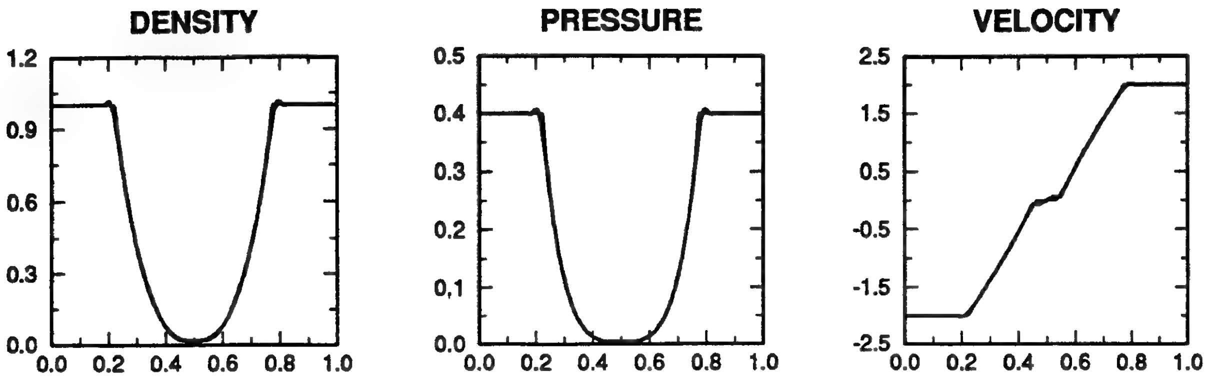


figure 6: Sjogreen test case: with no Min-Mod limitations

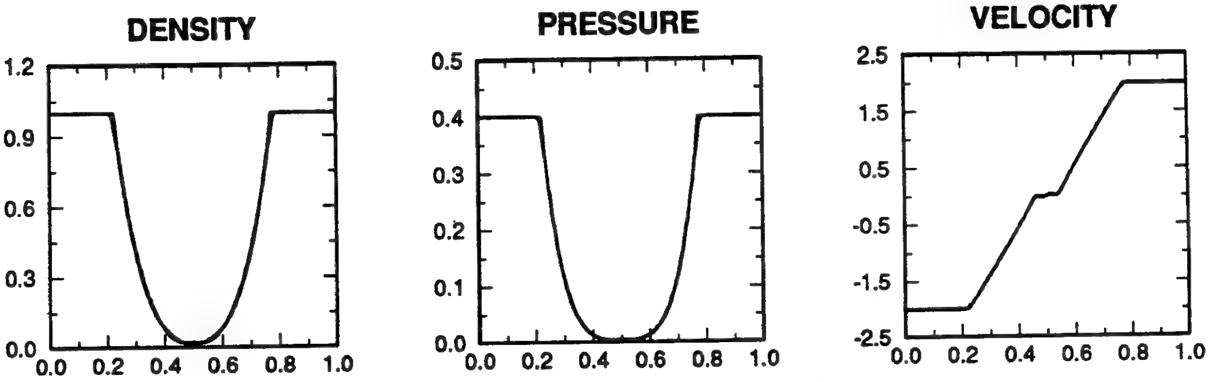


figure 7: Sjogreen test case: with Min-Mod limitations

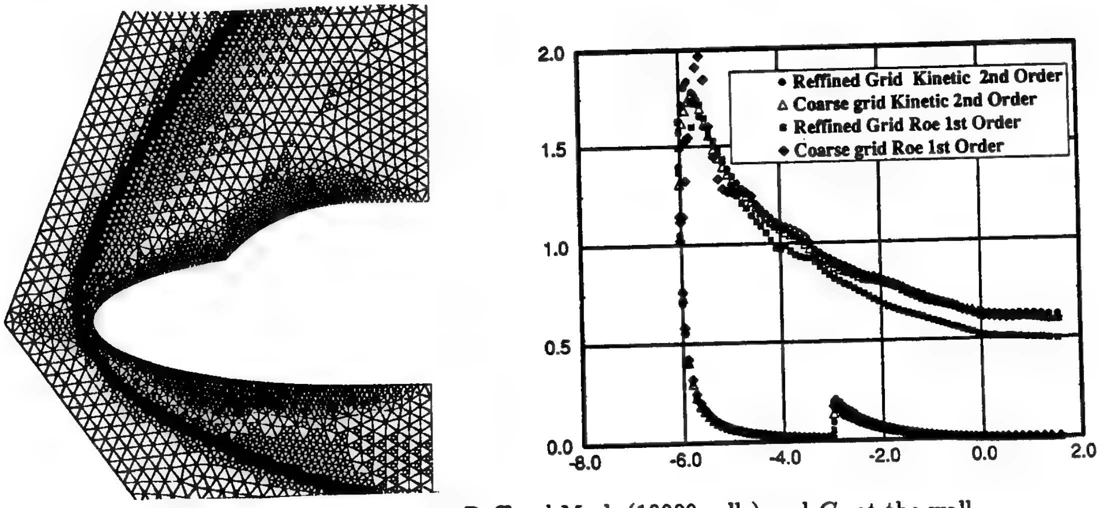
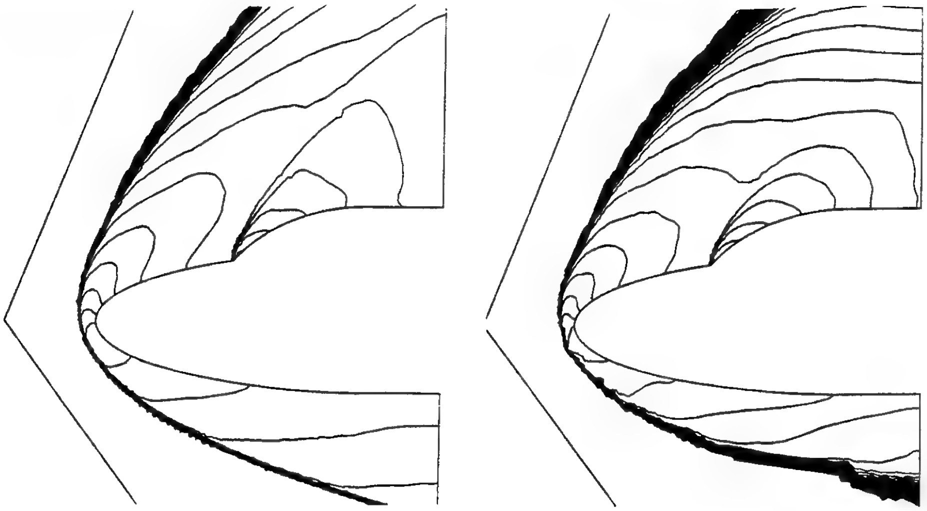


figure 8: Hermes test case: Reffined Mesh (10000 cells) and C_p at the wall



Kinetic Scheme

Roe scheme

figure 9: Hermes test case: Iso Mach Lines on the Reffined Mesh

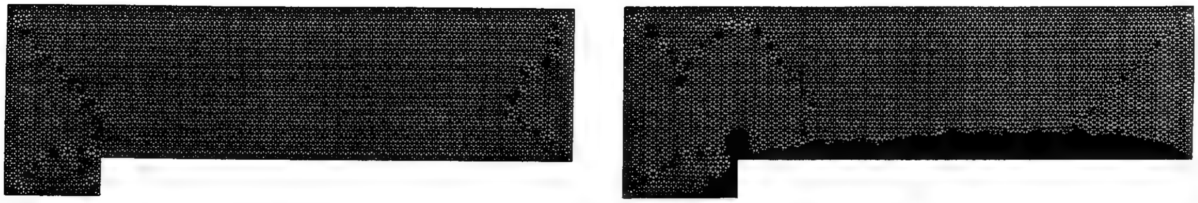
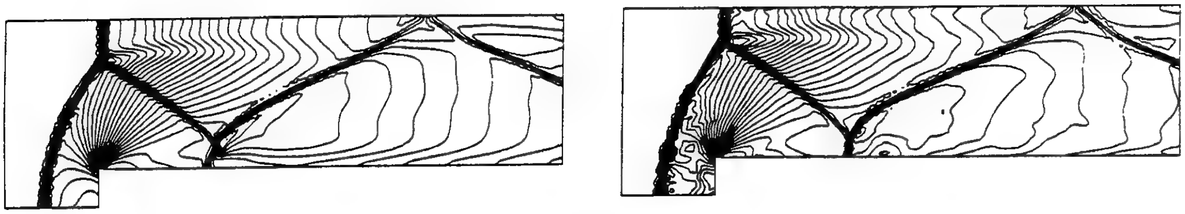


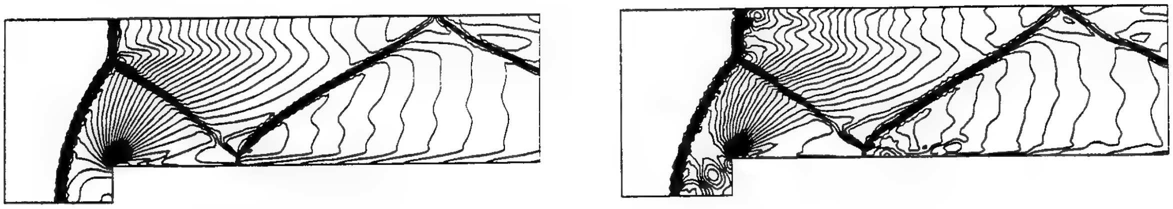
figure 10: Medium Mesh (9000 cells) and Reffined Mesh (14000 cells)



Kinetic Scheme

Roe scheme

figure 11: Emery test case: Iso density Lines on the medium Mesh



Kinetic Scheme

Roe scheme

figure 12: Emery test case: Iso density Lines on the reffined Mesh

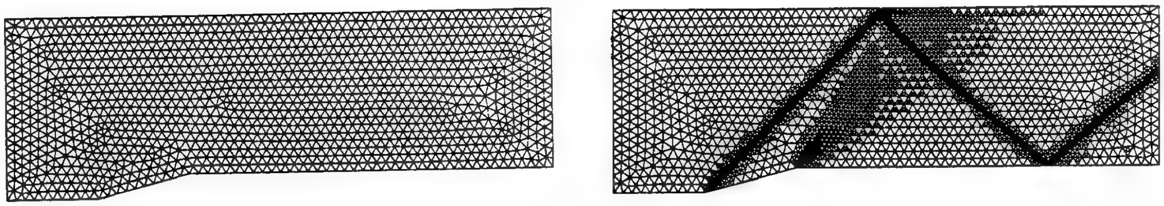
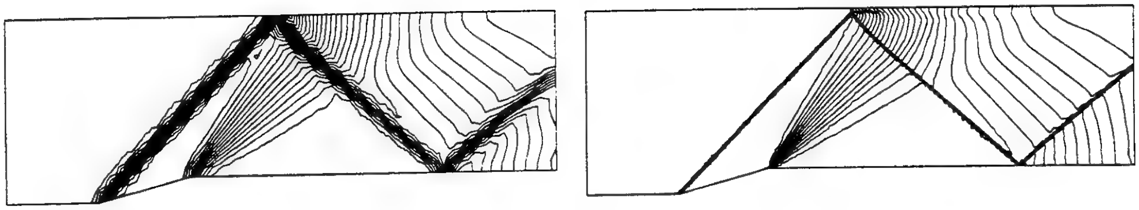


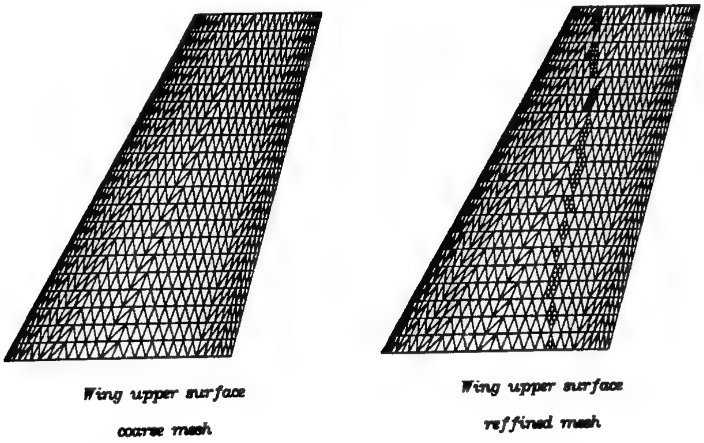
figure 13: Coarse Mesh (2500 cells) and Reffined Mesh (9000 cells)



Coarse mesh

Reffined mesh

figure 14: Iso density Lines



Wing upper surface
coarse mesh

Wing upper surface
reffined mesh

figure 15: Coarse Mesh and Reffined Mesh on the Wing

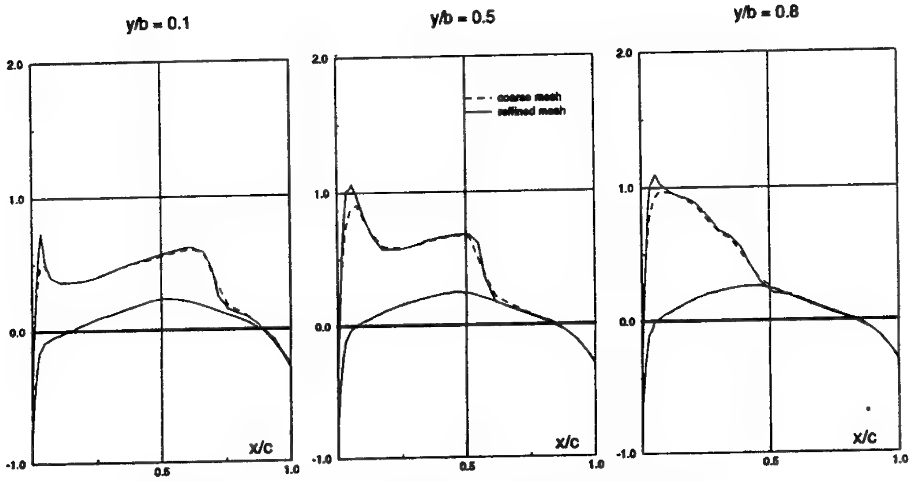
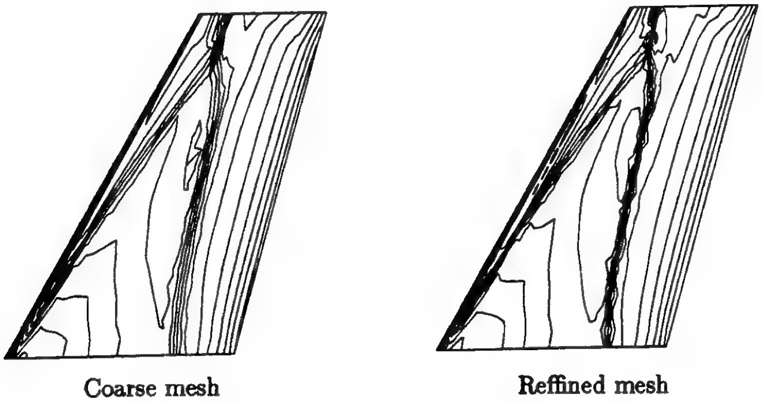


figure 16: C_p on the Wing at different sections



Coarse mesh

Reffined mesh

figure 17: Iso Pressure Lines on the Wing

A meshless technique for computer analysis of high speed flows

T. Fischer, E. Oñate and S. Idelsohn
International Center for Numerical
Methods in Engineering
Edificio C-1, Campus Norte UPC
C/ Gran Capitán, s/n
08034 Barcelona, Spain

1. ABSTRACT

This paper describes the results of the research carried out by the authors in the computer modelling of flow problems using an approximation based on "clouds of points" which does not require the definition of a mesh. The so called Finite Point Method (FPM) [5] is presented showing some examples for the solution of the 1D convection diffusion equation and 2D compressible inviscid flows.

2. INTRODUCTION

The finite element method (FEM) and the finite volume method (FVM) are well established numerical techniques whose main advantage is their ability to deal with complicated domains in a simple manner while maintaining a local character in the approximation. Both methods seek to divide the total domain into a finite number of subdomains (or elements) wherein a volume integration is performed. For these reasons the subdomains are limited by some regularity of geometrical conditions such as having a positive volume or a limited aspect ratio between elements, angles, etc. Although this poses no serious difficulties for 2D situations, the lack of efficient 3D mesh generators makes the solution of 3D problems a difficult task.

It is widely acknowledged that efficient 3D mesh generation remains one of the big challenges in FE and FV computations. Thus, even the more complex problems in CFD, such as some 3D solutions of the Navier-Stokes equations, can be accurately tackled nowadays providing an acceptable 3D mesh is available. However, the generation of 3D meshes, despite major recent advances is still a bottle neck and it can absorb far more time and effort than the numerical solution itself.

Different authors have recently investigated the possibility of deriving numerical methods without using meshes. Nayroles *et al* [1] proposed a technique, calling it diffuse element method (DEM), where only some nodes and a boundary description is necessary to formulate the Galerkin equations. The intepolating functions are polinomials fitted to the nodel values by a least squares approximation. Although no finite element mesh is explicitly required in this method, still some kind of "auxiliary grid" is needed in order to compute numerically the integral expressions deriving from the Galerkin approach. This requirement may prelude the successful extension of the DEM to 3D problems.

More recently, Belytschko *et al* [2] have proposed an extension of the DEM which they call the element-free Galerkin (EFG) method. In that work, generalized moving least squares interpolants typically exploited in curve and surface fitting are used to define the local approximation. This provides additional terms in the derivatives of the unknowns field omitted by Nayroles *et al* [1]. In addition, a regular cell structure is chosen as the "auxiliary grid" to compute the integrals by means of a higher order quadrature. Finally, Lagrange multipliers are used to enforce the essential boundary conditions. The same approach has been further generalized by Liu *et al* [3] by introducing concepts from wavelet theory.

The use of "clouds of points" to define local approximations is by no means new and it has enjoyed some popularity among finite difference (FD) practitioners to derive generalized FD schemes in arbitrary irregular grids. Here typically the concept of a "star" of nodes is introduced to derive FD approximations by means of a local Taylor expansion using the information by the number and position of nodes contained in each star. These ideas have been successfully applied in fluid mechanics under the name of

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

Smooth Particle Hydrodynamics Method. A recent extension of these concepts to the solution of high speed flow problems has recently been attempted by Batina [4].

In this paper a general methodology for the numerical solution of high speed flows using a finite set of arbitrary points is described. The approach proposed incorporates the main features of generalized finite difference schemes and other more recent point data based procedures such as the DEM and the EFG [5]. The theoretical basis of the method in the context of the solution of viscous and inviscid flows are described in some detail. The accuracy and applicability of this method is shown in some examples of application in 1D and 2D flow problems.

3. METHODOLOGY

3.1 The Finite Point method (FPM)

From a polynomial expansion of order m a function $u(x)$ can be approximated in a local interpolating domain Ω_i (sometimes also termed "clouds")

$$u(x) \simeq \hat{u}(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + ... + \alpha_m x^{m-1} =$$

$$= \mathbf{p}^T(x)\boldsymbol{\alpha} \qquad (1)$$

where the base functions $\mathbf{p}^T = [1, x]$ for $m = 2$ and $\mathbf{p}^T = [1, x, x^2]$ for $m = 3$ in one dimension [5].

The above approximation can now be sampled at n points within Ω_i where the values of the unknown $u_i^h = u(x_i)$ are sought, i.e.

$$\mathbf{u}_i^h = \left\{ \begin{array}{c} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ . \\ . \\ \mathbf{p}_n^T \end{array} \right\} \boldsymbol{\alpha} = \mathbf{C}\boldsymbol{\alpha} \qquad (2)$$

where $\mathbf{C}$ is a nxm matrix.

3.1.1 *Finite element interpretation*

If $n = m$, a standard finite element interpolation is obtained by inverting eq. (2) and substituting into (1) as

$$\hat{u}(x) = \mathbf{p}^T \mathbf{C}^{-1} \mathbf{u}^h = \mathbf{N}^T \mathbf{u}^h \qquad (3)$$

$\mathbf{N}$ being the standard finite element shape functions [7].

3.1.2 *Least squares interpolation*

Increasing the number of nodes in Ω_i to $n > m$, we cannot directly invert $\mathbf{C}$ anymore. However, through least squares approximation a square matrix is obtained which can be inverted if $\mathbf{C}$ has a full rank, which is assumed in what follows. Hence, the following sum of squares can be written using eq. (1)

$$J = \sum_{j=1}^{n}(u_j^h - \hat{u}(x_j))^2 = \sum_{j=1}^{n}(u_j^h - \mathbf{p}^T\boldsymbol{\alpha})^2 \qquad (4)$$

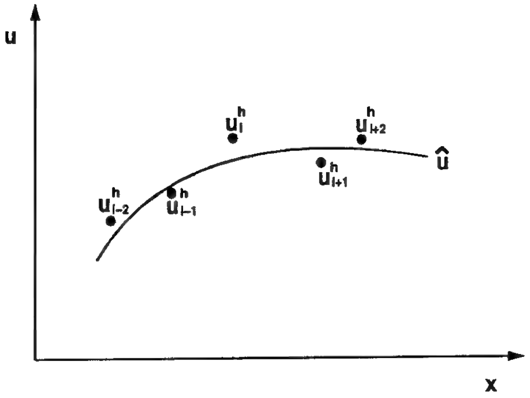


Figure 1: Nodal unknowns u and the interpolated function $\hat{u}$.

Minimizing J with respect to α, $\frac{\partial J}{\partial \alpha} = 0$, yields

$$\boldsymbol{\alpha} = \mathbf{A}^{-1}\mathbf{B}\mathbf{u}^h \qquad (5)$$

with $\mathbf{A} = (\mathbf{C}^T\mathbf{C})$ and $\mathbf{B} = \mathbf{C}^T$. The new shape functions are now obtained as

$$\mathbf{N}^T = \mathbf{p}^T\mathbf{A}^{-1}\mathbf{B} \qquad (6)$$

This means that an interpolated curve $\hat{u}(x)$ is generated from some point values u_j^h in each cloud as shown in Figure 1. Note that the fitted curve does not necessarily pass through the nodal unknowns u_j^h.

Recently, Batina [4] has used a similar type of least squares fit for fluxes and stresses in compressible flow analysis. However, he avoids the direct inversion of matrix $\mathbf{A}$ by doing a QR decomposition.

In the present approach the danger of $\mathbf{A}$ being singular is avoided by appropriately selecting the points in the interpolation domain Ω_i . This reduces both computational cost and memory.

3.1.3 *Weighted Least Squares Approach*

A drawback of the interpolation procedure so far presented is that equal weight is given to all the points in Ω_i. This can rapidly cause a deterioration of the approximation [6]. A remedy can be the introduction of weighting functions, such as a Gauss function, which will be described next.

Following the least squares approach from above, we can directly include the weighting functions $w(x_j)$ in eq. (4):

$$J = \sum_{j=1}^{n} w(x_j)(u_j^h - \hat{u}(x_j))^2 = \sum_{j=1}^{n} w(x_j)(u_j^h - \mathbf{p}^T\alpha)^2 \tag{7}$$

Again minimizing J with respect to α, we obtain

$$\boldsymbol{\alpha} \; = \; \mathbf{A}^{-1}\mathbf{B}\mathbf{u}^h \tag{8}$$

with $\mathbf{A} = w(x_j)(\mathbf{C}^T\mathbf{C})$ and $\mathbf{B} = \mathbf{C}^T\mathbf{W}$. $\mathbf{W}$ is now a diagonal matrix containing the weights $w(x_j)$ at each point in Ω_i.

In [6], the authors demonstrate a strong sensitivity to the number of points chosen within each cloud Ω_i if no weighting functions are used. In an example, the shape function plots show a drastic deterioration for both linear and quadratic base functions $\mathbf{p}$.

3.2 The FPM in a one dimensional context

Let us now apply the theoretical background to a typical test problem, the linear 1D convection diffusion equation, and compare its results to known solutions from the FEM. Consider the 1D convection-diffusion equation:

$$\frac{\partial u}{\partial t} + A\frac{\partial u}{\partial x} - \frac{\partial}{\partial x}\left(\kappa\frac{\partial u}{\partial x}\right) = 0 \quad \text{in } \Omega \tag{9}$$

with $u = u(t,x)$ in Ω; $u(t,0) = u_0$ in Γ_0, $u(t,L) = u_L$ in Γ_L and $\Gamma = \Gamma_0 \cup \Gamma_L$.

At steady state ($\frac{\partial u}{\partial t} = 0$) equation (9) becomes:

$$A\frac{\partial u}{\partial x} - \frac{\partial}{\partial x}\left(\kappa\frac{\partial u}{\partial x}\right) = 0 \tag{10}$$

with u=u(x).

Taking A and κ constant, the analytical solution of this first order homogeneous differential equation is obtained as:

$$u(x) = u_0 + (u_L - u_0)\frac{1 - e^{\frac{A}{\kappa}x}}{1 - e^{\frac{A}{\kappa}}} \tag{11}$$

With $u_0 = 1$ and $u_L = 0$, equation (11) reduces to

$$u(x) = 1 - \frac{1 - e^{\frac{A}{\kappa}x}}{1 - e^{\frac{A}{\kappa}}} \tag{12}$$

A test for time marching schemes is solving equation (9) by iterating until steady state is reached to approximate the exact result. This is usually done by expanding equation (9) in time using a Taylor series:

$$u^{n+1} = u^n + \sum_{i=1}^{\infty}\frac{\Delta t^i}{i!}\frac{\partial^i u}{\partial t^i} \tag{13}$$

A discretization in space must be performed next. First, known and proven finite element methods will be presented, and then the finite point method proposed will be described.

3.2.1 *Finite element solution*

It is well known that the exact solution to this problem can be nodally reproduced by the finite element method using the following Petrov-Galerkin methods for all ranges of the Peclet number Pe [7]. This can be achieved by expanding equation (13) up to first order, replacing $\frac{\partial u}{\partial t}$ with equation (9) and discretizing in space using Petrov-Galerkin shape functions:

$$W = N + \frac{h}{2}\alpha_{opt}\frac{\partial N}{\partial x} \tag{14}$$

with the upwind parameter $\alpha_{opt} = coth(Pe) - \frac{1}{Pe}$ which is optimal for this equation. The so called Taylor-Galerkin approach can also be used to recover exact nodal values for this problem. In fact, if equation (13) is expanded up to second order (omitting third order derivatives) and standard Galerkin linear shape functions are used, equivalence to the Petrov-Galerkin scheme can be proved [7] for

$$\Delta t = \Delta t_{opt} = \frac{h}{A}\alpha_{opt} \tag{15}$$

$$\text{and } Pe = \frac{Ah}{2\kappa} \tag{16}$$

Figure 2 shows the exact nodal values obtained using a Taylor-Galerkin two-step scheme with Δt_{opt} and $Pe = 1$ [7,8].

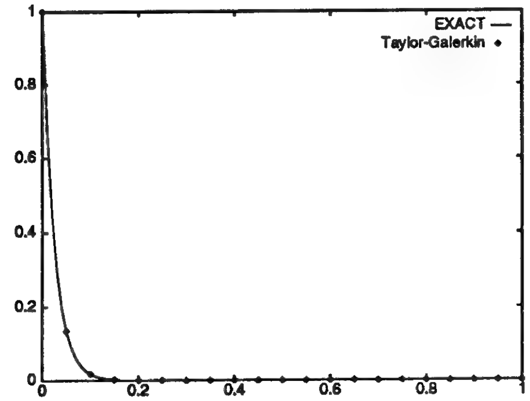


Figure 2: Exact solution to the convection diffusion equation using a Taylor-Galerkin scheme.

3.2.2 *The finite point method (FPM)*

Let us now analyze the finite point method in the context of the 1D convection-diffusion equation. Integrating eq. (9) in time, performing a Taylor expansion of eq. (13) up to second order leads to:

$$u^{n+1} = u^n + \Delta t \frac{\partial u}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial t^2} \tag{17}$$

inserting eq. (9) into (17) and omitting third order derivatives, we obtain:

$$u^{n+1} = u^n - \Delta t \left[\left(A \frac{\partial u}{\partial x} - \frac{\partial}{\partial x} (\kappa \frac{\partial u}{\partial x}) \right) - \frac{A^2 \Delta t}{2} \frac{\partial^2 u}{\partial x^2} \right] \tag{18}$$

The discretization of the computational domain is performed locally using arbitrary points, without the need for fixed connectivities in a conventional mesh. Performing a least squares approximation in the vicinity of a point using using n points, we obtain an estimation of the necessary spatial derivatives $\frac{\partial u}{\partial x}$ and $\frac{\partial^2 u}{\partial x^2}$. Substituting these derivatives into eq. (18) leads to a system of equationsfrom which the unknown point values $u(_j$ can be found for each time increment. The approach is equivalent to using a point collocation scheme [5].

As explained earlier, the unknown functions $u(x)$ and its derivatives may be expanded as follows in a given cloud, for the linear case (in what follows we assume $u(x) = \hat{u}(x)$):

$$u(x) = \alpha_1 + \alpha_2 x = \mathbf{p}^T \boldsymbol{\alpha}$$

$$\frac{\partial u}{\partial x} = \alpha_2 \tag{19}$$

and for the quadratic case:

$$u(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 = \mathbf{p}^T \boldsymbol{\alpha}$$

$$\frac{\partial u}{\partial x} = \alpha_2 + 2\alpha_3 x \tag{20}$$

$$\frac{\partial^2 u}{\partial x^2} = 2\alpha_3$$

For the linear case, it is not possible to directly compute the necessary second order derivatives. This can be overcome by performing an accumulation of differences at the central point and the rest of the points j within the cloud. Hence,

$$\frac{\partial^2 u}{\partial x^2} = \sum_{j=2}^{n} \frac{(u_j - u_1)}{h^2} \tag{21}$$

with point 1 being the central point.

It can be shown that this scheme, for equally spaced points ($n = 3$ and $m = 2, 3$), is equivalent to central-differences which in turn is equivalent to FEM with Galerkin shape functions [8].

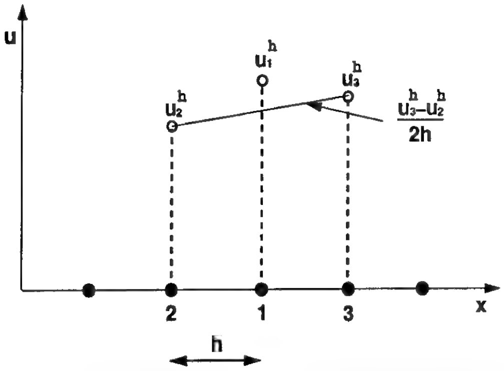


Figure 3: Equally spaced points and their domain of influence for $n = 3$. Note the derivative is equivalent to a central difference approximation.

The following shows this for the linear case (m=2). Consider three points (1,2,3) with the coordinates

(x_1,x_2,x_3) and at equal distance h from each other as shown in Figure 3. Let their unknown values be u_1, u_2, u_3, respectively. Then, **A** from equation (5) can be calculated as

$$\mathbf{A} = \begin{bmatrix} n & \sum_{j=1}^{n} x_i \\ \sum_{j=1}^{n} x_i & \sum_{j=1}^{n} x_i^2 \end{bmatrix} = \begin{bmatrix} 3 & 3x_1 \\ 3x_1 & 3x_1^2 + 2h^2 \end{bmatrix} \tag{22}$$

Inversion of **A** leads to

$$\mathbf{A}^{-1} = \begin{bmatrix} \frac{1}{3} + \frac{x_1^2}{2h^2} & -\frac{x_1}{2h^2} \\ -\frac{x_1}{2h^2} & \frac{1}{2h^2} \end{bmatrix} \tag{23}$$

Eq. (5) gives the polynomial coefficients α_1 and α_2 as:

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \mathbf{A}^{-1}\mathbf{B}\mathbf{u} \tag{24}$$

The first derivatives in the linear case are constant. Using eq. (19), (23) and (24) they become now:

$$\frac{\partial u}{\partial x} = \alpha_2 = (\frac{x_1}{2h^2} - \frac{x_1}{2h^2})u_1 + (\frac{x_1}{2h^2} - \frac{x_1 - h}{2h^2})u_2 +$$

$$+(\frac{x_1}{2h^2} - \frac{x_1 + h}{2h^2})u_3 = \frac{u_3 - u_2}{2h} \tag{25}$$

which is exactly a central difference approximation. The second differences $\frac{\partial^2 u}{\partial x^2}$ are taken as accumulated differences at the central node 1, which leads to

$$\frac{\partial^2 u}{\partial x^2} \simeq \frac{u_2 - u_1}{h^2} + \frac{u_3 - u_1}{h^2} = \frac{u_2 - 2u_1 + u3}{h^2} \tag{26}$$

which also is a central difference approximation. By analogy, we can derive similar statements for the quadratic case ($m = 3$).

Having shown the equivalence of FPM ($n = 3$, $m = 2, 3$) with FEM, it should be possible to recover nodally excact values for $n = 3$ and $m = 2, 3$ using the finite point method. Figure 4 demonstrates this for $Pe = 1$.

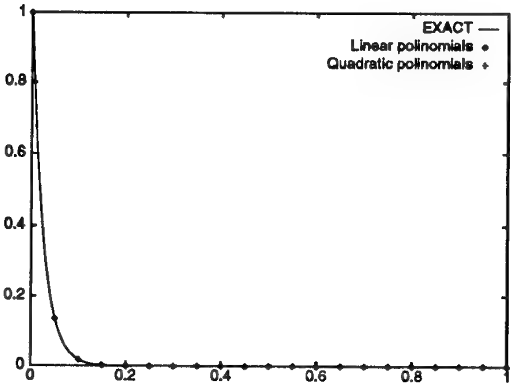


Figure 4: Exact nodal values obtained by FPM for $Pe = 1$, $n = 3$ and $m = 2, 3$.

However, if the number of points in the local interpolating domain Ω_i increases ($n > 3$), the algorithm introduces excessive diffusion and the quality of the result deteriorates, especially near strong gradients.

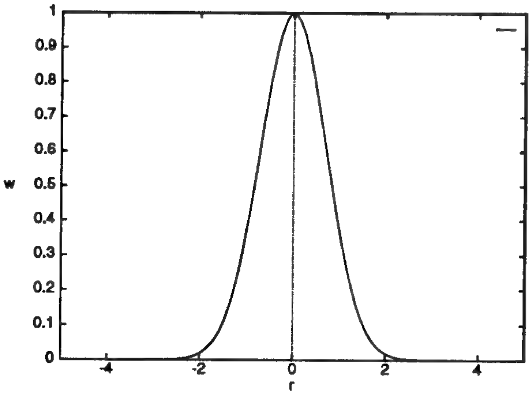


Figure 5: Gauss weighting function; $w(r_j)$ quickly decreases as r_j increases (for $\lambda_i c = 1$).

3.2.3 *Introduction of weighting functions*

It is now of interest to see if the results can be improved by the use of weighting functions w_j within Ω_i. The idea is to give additional weight to points close to the central point and reduce it for points farther away. Within each Ω_i we define w_j as a function of the distance of each point j to the central point:

$$w_j = w(r_j), \text{ and } r_j = |x_j - x_1| \tag{27}$$

A possible choice for weighting could be a Gauss distribution which was also used in all following calculations:

$$w(r_j) = e^{-\left(\frac{r_j}{\lambda_i c}\right)^p} \tag{28}$$

where λ_i is a characteristic length in each cloud Ω_i. c and p are user defined constants to adjust the sensitivity of the weighting function. Usually, $c = 1$ and $p = 2$ are chosen. Figure 5 displays $w(r_j)$ graphically for $\lambda_i c = 1$. Further information on the FPM can be found in [5,6].

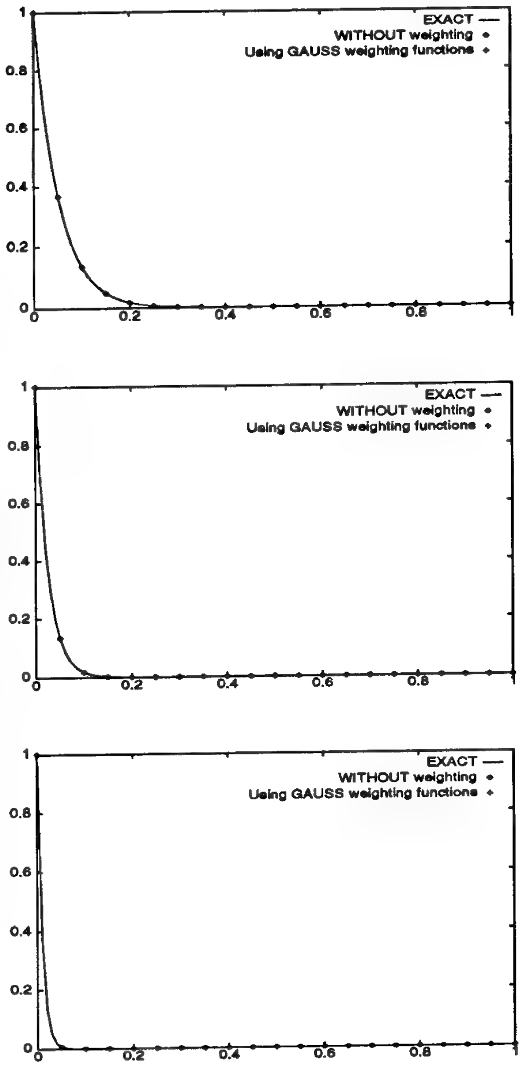


Figure 6: Convection diffusion equation using linear base functions and 3 points per cloud for a) $Pe = 0.5$, b) $Pe = 1$ and c) $Pe = 2.5$.

4. NUMERICAL EXAMPLES

4.1 1D Convection diffusion equation

Let us test the solution of equation (18) using the FPM without weighting and with Gaussian weighting functions.

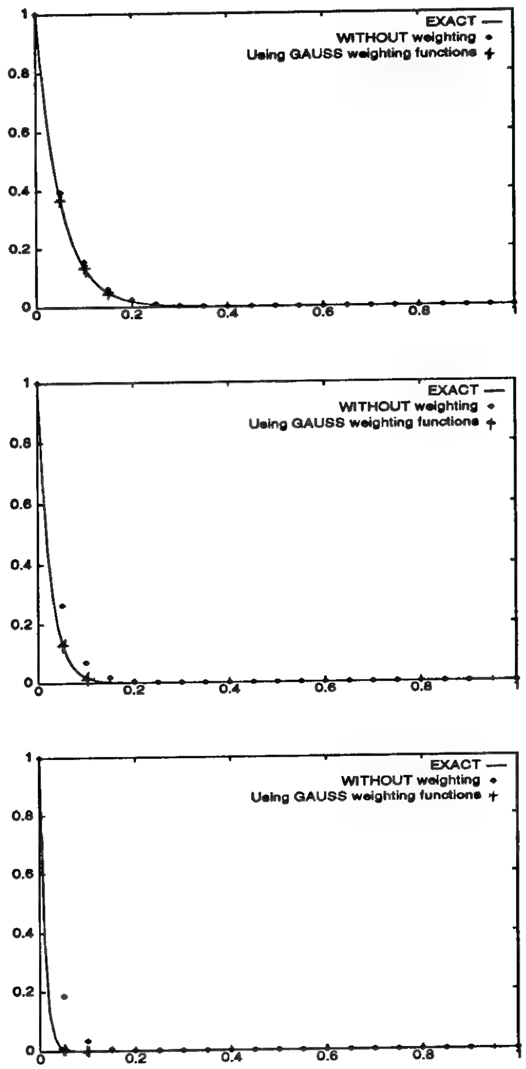


Figure 7: Convection diffusion equation using linear base functions and 4 nodes per cloud for a) $Pe = 0.5$, b) $Pe = 1$ and c) $Pe = 2.5$.

Fig. 6 shows the FP solution using linear base functions ($m = 2$) and 3 points per cloud for three different Peclet numbers: a) $Pe = 0.5$ b) $Pe = 1.0$ and c) $Pe = 2.5$. Observe that exact nodal values are obtained in all cases.

However, as the number of nodes in the cloud is increased, a deterioration of the solution is visible when no weighting functions are employed. Figures 7 and 8 demonstrate this behavior for $n = 4$ and $n = 5$.

Using Gaussian weighting functions, the improvement of the solution is impressive. With $\lambda_i = r_{min}$, where r_{min} refers to the minimum distance of r in Ω_i, practically exact nodal values are recovered for this 1D test problem (see Figures 7 and 8).

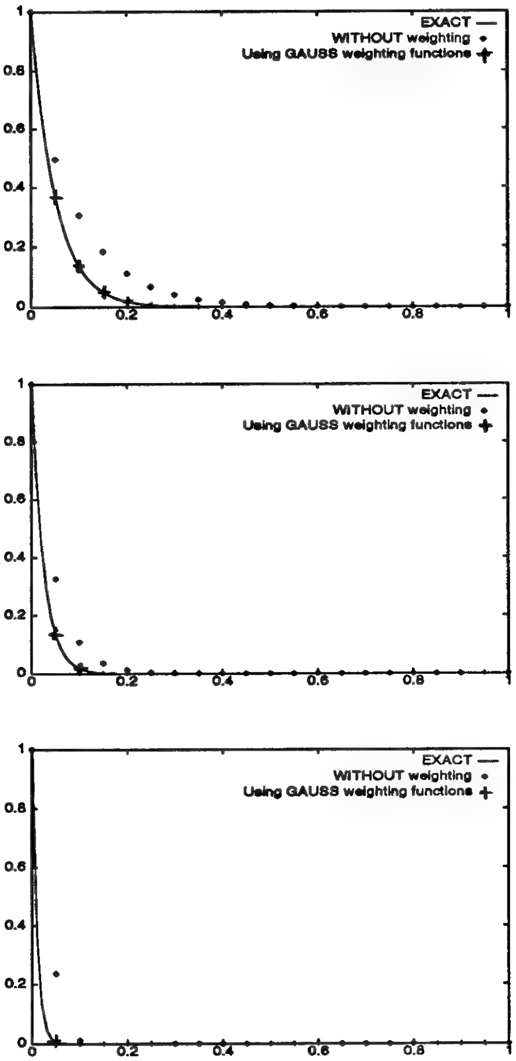


Figure 8: Convection diffusion equation using linear base functions and 5 nodes per cloud for a) $Pe = 0.5$, b) $Pe = 1$ and c) $Pe = 2.5$.

The extension to quadratic base functions ($m = 3$) exhibits a more drastic need for using of weighting functions. Whereas with 3 noded clouds ($n = 3$) exact nodal values are computed (Figure 9), strong oscillations occur as n is increased (Figure 10). These oscillations disappear if a weighting interpolation is used.

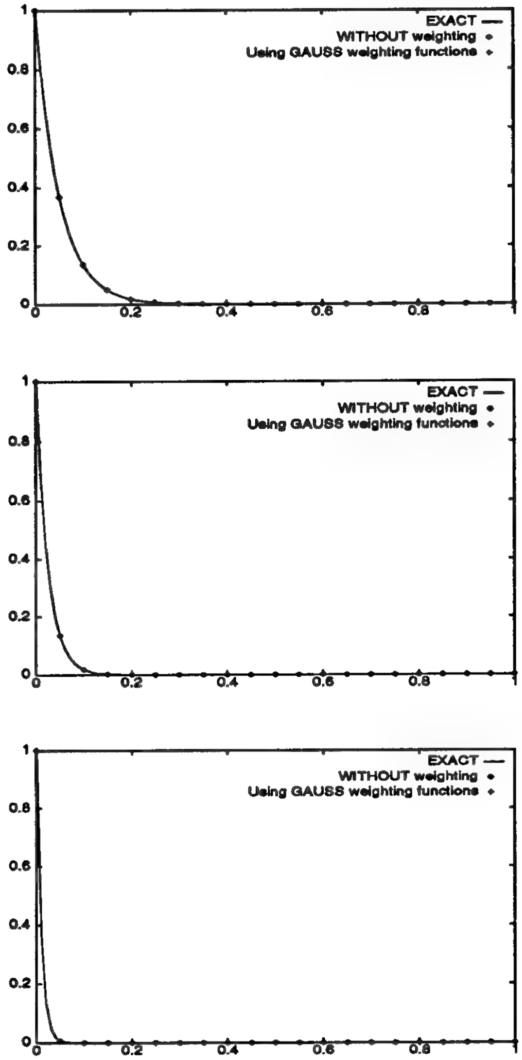


Figure 9: Convection diffusion equation: quadratic base functions and 3 nodes per cloud for a) $Pe = 0.5$, b) $Pe = 1$ and c) $Pe = 2.5$.

Additionally, we have also found that the quality of the results worsens as the Peclet number increases if no weighting functions are employed. Note that again nearly exact nodal solutions are recoverd by employing Gaussian weighted interpolation.

4.2 Extension to the 2D Euler equations

4.2.1 *Governing equations*

The ideas from the one dimensional problem are extended to the solutions of the non linear two dimensional Euler equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}_k}{\partial x_k} = 0 \quad j = 1,2 \tag{29}$$

where

$$\mathbf{u} = \left\{ \begin{array}{c} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho \epsilon \end{array} \right\} \text{ and } \mathbf{f}_k = \left\{ \begin{array}{c} \rho u_k \\ \rho u_k u_1 + p\delta_{1k} \\ \rho u_k u_2 + p\delta_{2k} \\ (\rho\epsilon + p)u_k \end{array} \right\}$$

The different terms have the usual meaning [8].

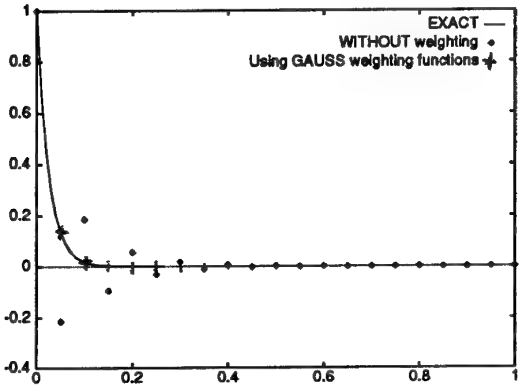


Figure 10: Convection diffusion equation using quadratic base functions and 5 points per cloud for $Pe = 1$.

4.2.2 *Time discretization*

A two-step scheme is employed in order to advance the solution in time towards steady state, i.e.

$$\mathbf{u}^{n+\frac{1}{2}} = \mathbf{u}^n - \frac{\Delta t}{2}\frac{\partial \mathbf{f}_k}{\partial x_k}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \Delta t \frac{\partial \mathbf{f}_k^{n+\frac{1}{2}}}{\partial x_k} \simeq \mathbf{u}^n - \Delta t \frac{\partial}{\partial x_k}\left(\mathbf{f}_k(\mathbf{u}^{n+\frac{1}{2}})\right) \tag{30}$$

4.2.3 *Stability*

The two-step scheme leads to a conditionally stable explicit second order algorithm of with the following limit for Δt:

$$\Delta t = \frac{Ch}{|\mathbf{v}| + c} \text{ and } C \leq 1 \tag{31}$$

where C is the Courant number.

A difficulty in a multidimensional context arises from the determination of h in a given cloud of points Ω_i. In finite elements, using linear triangular elements, h is defined according to the minimum height within each element [8]. In meshless methods, a clear definition has not been presented yet. In our work, h has been taken equal to λ_{min}, this being the minimum distance to the center point within each interpolation domain Ω_i.

4.2.4 *Balancing dissipation*

Since the hyperbolic Euler equations do not contain any diffusion terms, some balancing damping must be added to prevent unphysical oscillations. Following Jameson [9], 2nd and 4th order diffusion terms are added to the fluxes. These terms are constructed as follows in the FPM:

$$\mathbf{d_i} = -(|\mathbf{v}| + c)\sum_{j=2}^{n}(\mathbf{e}_j - \mathbf{e}_i)\frac{w_j}{\sum_{j=2}^{n} w_j}$$

$$\mathbf{e_i} = \epsilon_i^{(2)}\mathbf{u}_i - \epsilon_i^{(4)}\sum_{j=2}^{n}(\mathbf{u}_j - \mathbf{u}_i) \tag{32}$$

where w_j are the same weighting functions used in the interpolations of eq. (8) and the coefficients of eq. (32) are obtained as:

$$\epsilon_i^{(2)} = \alpha^{(2)}\frac{\sum_{j=2}^{n}|p_j - p_i|}{\sum_{j=2}^{n} p_j + p_i}\ ; \ \epsilon_i^{(4)} = \max(0, \alpha^{(4)} - \epsilon_i^{(2)}) \tag{33}$$

$\alpha^{(2)}$ and $\alpha^{(4)}$ are user defined constants. The summation j extends accross the number of points in each cloud and is accumulated at both the central point i and the point j. In subcritical flows $\epsilon_i^{(2)}$ is generally switched off.

4.2.5 *Selection of points*

In a multidimensional domain, the difficulty arises on how to define each local interpolating domain. Even though weighting functions are employed, it is still necessary to choose the most significant points for each Ω_i. For the results of this paper, the central point plus the $n-1$ closest points are chosen. However, a condition of quadrants is imposed such that there must be at least one point in every quadrant of orthogonal axes. This leads to a minimum of 5 points per cloud.

At the boundary, the two points adjacent to the central point on the boundary plus the closest points are chosen. Another condition is that no boundary section is crossed so that points from the opposite side are not chosen. For instance, at the trailing edge of an airfoil, the closest points to a point on one side of the airfoil may lie accross the wall on the other side. Since there is a physical separation of these points, they are not included in the same interpolation domain.

4.3 2D Results

4.3.1 *Subsonic test case*

The first 2D test case considered is a NACA0012 profile with a free stream Mach number of 0.5 and 0 degrees angle of attack, analyzed by Zienkiewicz *et al* [10,11]. In order to compare solutions, a finite element solution has been taken for comparison. The meshless grid of 2556 points is shown in Figure 11. The same points have been used for the FE solution on the equivalent unstructured triangular mesh obtained using a standard advancing front technique [7,8].

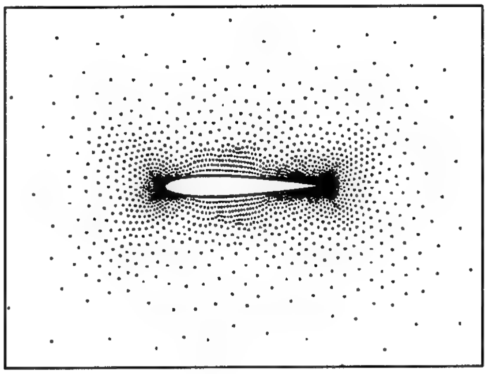


Figure 11: Point distribution around a NACA0012 profile

Again, the idea is to compare the influence of the weighting functions in the finite point approximation. In previous reports [5,6] we have shown results proving the superiority of weighting functions in a 2D context, but without using weighting functions for the balancing diffusion terms (see eq. (32)). Here, the benefit of the weighted diffusion terms shall be presented.

The results of the FPM were obtained by employing 7 nodes in Ω_i, $\lambda = \lambda_{min}$, $c = 1$ and linear base functions ($m = 3$). A global comparison of the meshless solution is shown in Figure 12. In a), b), c) and d) the mesh, the Taylor-Galerkin solution, a four-stage Runge-Kutta Galerkin result and the FPM solution for the density are presented, respectively. Qualitatively, all results are very similar. In Figure 13 close-up views in the stagnation area enhance the comparison of density contours of a) FPM without weighted diffusion, b) FPM with full Gauss weighting, c) RK-Galerkin and d) Taylor Galerkin. Note the improvement of solution b) with respect to a), not exhibiting any oscillations in the stagnation area through the use of weighted diffusion terms.

In Figure 14, a) velocity contours and b) velocity vectors in the stagnation zone are displayed, respectively.

4.3.2*Supersonic test case*

The second 2D test case is a hypersonic inviscid flow of Mach 8.15 around a double ellipse, which is well documented by the proceedings of the workshop in Antibes, 1991 [12]. The flow enters at an angle of 30 degrees. The solution is characterized by a strong primary bow shock and a weaker canopy shock.

To solve this problem, a grid of approximately 11000 points was generated using again the advancing front technique. Linear base functions (m=3) and 6 point clouds with Gaussian weighting were used. The residuals of the solution have been reduced to six orders of magnitude. Figures 15 a). b) and c) present the meshless grid, Mach number contours and density lines, respectively. Note that the solution is very smooth and the location of the shock is well captured. The numerical overshoot of about 3% in Mach number is within reasonable limits and it could be improved by increasing the balancing diffusion. The convergence of this solution was slow due to a low Courant number of 0.25 (avoiding neg. pressures).

Figure 16 a) demonstrates the high quality of the solution in the vicinity of the stagnation area showing no oscillations. Figure 16 b) displays the pressure coefficient c_p on the boundary of the double ellipse which compares well to other contributors [12].

5. REFERENCES

[1] Nayroles, B., Touzot, G. and Villon, P. "Generalizing the Finite Element Method: Diffuse Approximation and Diffuse Elements", Computational Mechanics, 10, 307-318, 1992

[2] Belytschko, T., Lu, Y. and Gu, L. "Element Free

Galerkin Methods", Int. Journal for Numerical Methods in Engineering, 37,229-256, 1994

[3] Liu, W.K., Jan, S. and Belytschko "Reproducing Kernel Particle Methods", Int. Journal for Numerical Methods in Engineering (to be published)

[4] Batina J., "A Gridless Euler/Navier Stokes Solution Algorithm for Complex Aircraft Applications", AIAA paper, 93-0333, Reno NV, January 11-14, 1993

[5] Oñate E., Idelsohn S. and Zienkiewicz O.C. "Finite Point Methods in Computational Mechanics", Publication CIMNE No. 67, July 1995

[6] Oñate E., Idelsohn S., Fischer T., Zienkiewicz O.C. "A Finite Point Method for analysis of fluid flow problems", 9th Int. Conf. on Finite Elements in Fluids, Venezia, Italy, 15-21 October 1995

[7] Zienkiewicz, O.C. and Taylor R.L. "The Finite Element Method", 4th Edition, Volume 2, Mc. Graw Hill, 1991

[8] Peraire, J. "A Finite Element Method for Convection Dominated Flows", PhD Thesis, University College of Swansea, 1986

[9] Jameson, A., Schmidt, W., and Turkel, E. "Numerical simulation of the Euler equations by finite volume methods usirg Runge-Kutta time stepping schemes" AIAA paper 81-1259. AIAA 5th Computational Fluid Dynamics Conf., 1981

[10] Zienkiewicz, O.C. and Wu, J. "A General Explicit or Semi-Explicit Algorithm for Compressible and Incompressible Flows", Inst. of Num. Meth. in Eng., University College of Swansea, CR/682/91, 1991

[11] Zienkiewicz, O.C., Codina, R., Morgan, K. and Sai, S. "A General Algorithm for Compressible and Incompressible Flow", Inst. of Num. Meth. in Eng., University College of Swansea, CR/842/94, June 1994

[12] Problem 6 of the Workshop on Hypersonic Flows for Reentry Problems, Antibes, France, January 22-25 1990

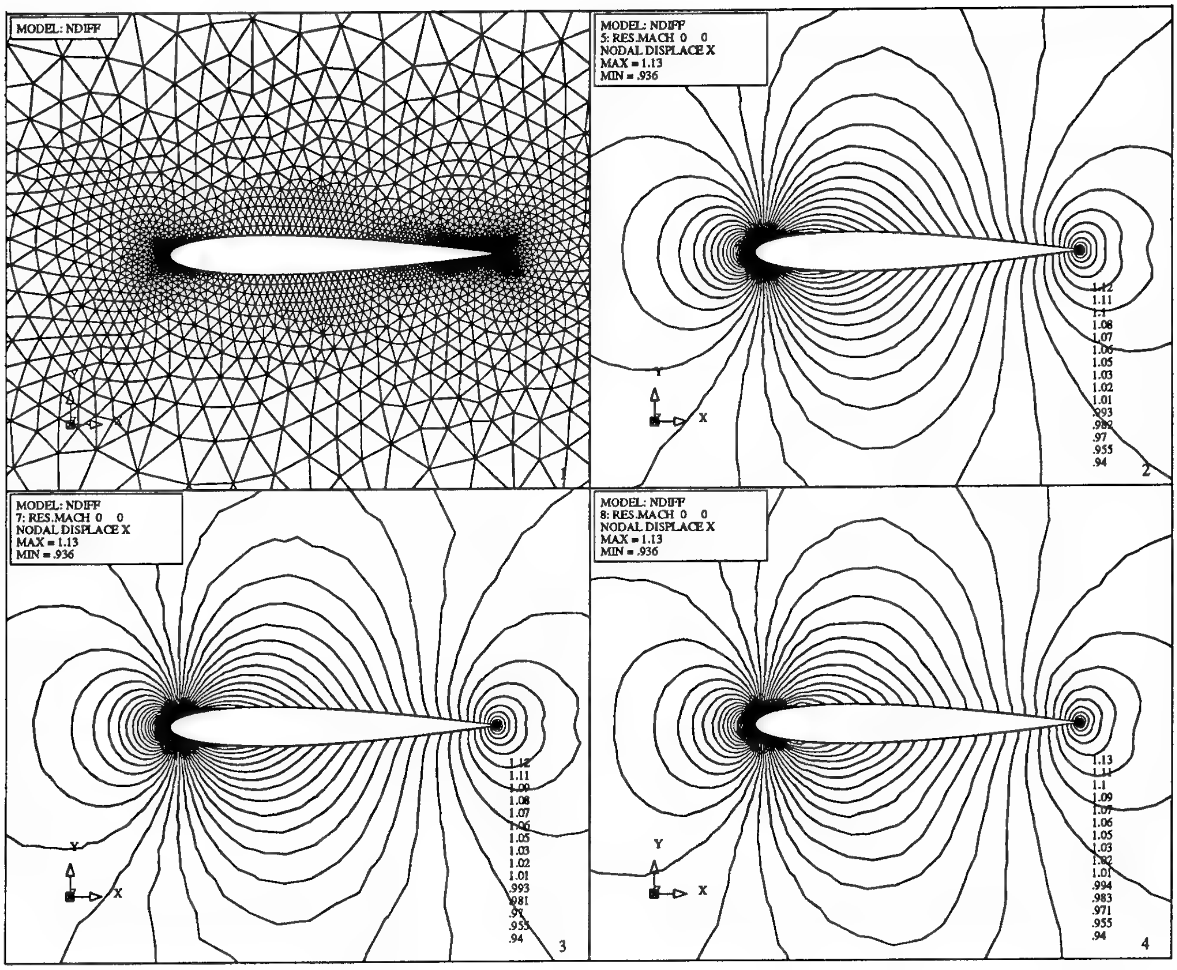


Figure 12: NACA0012 profile: a) Mesh, b) TG solution, c) RK solution and d) FPM solution are shown.

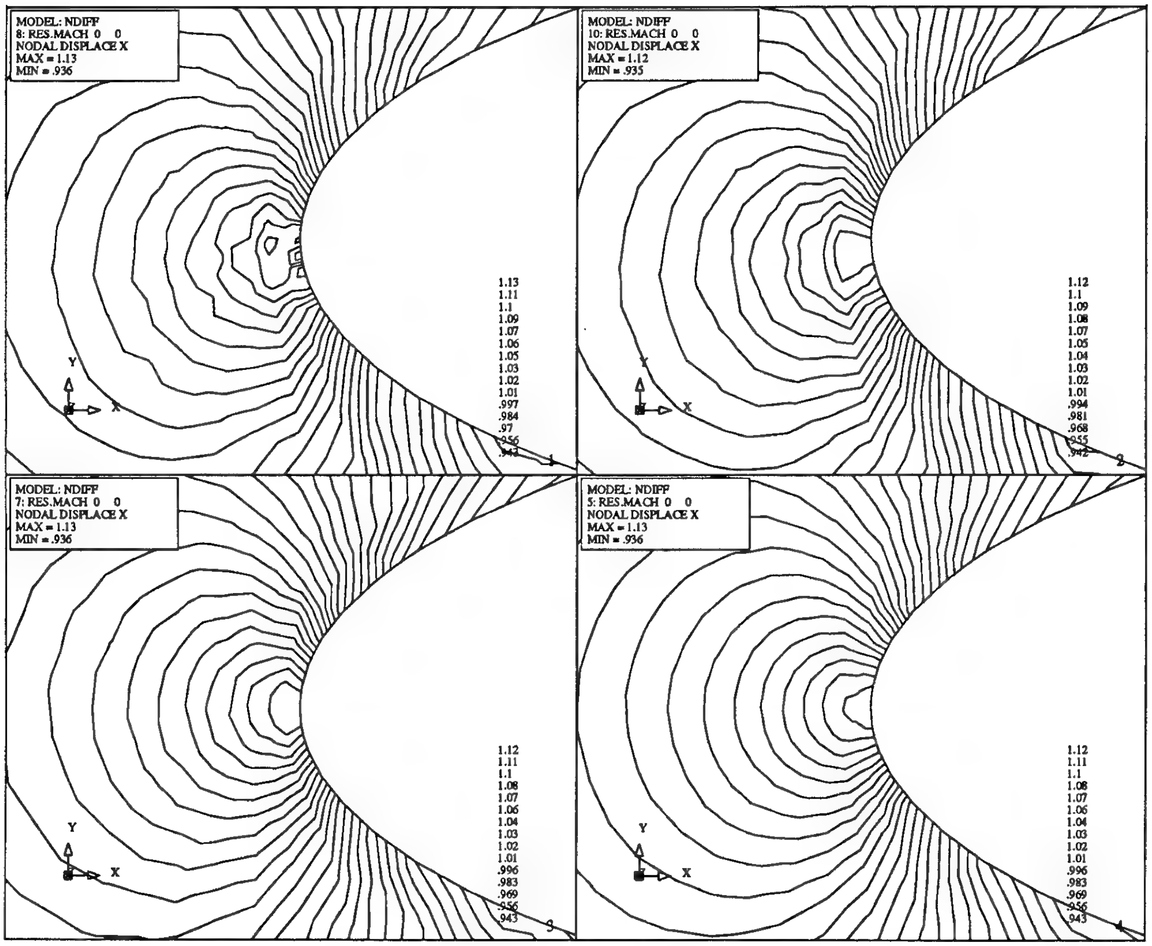


Figure 13: NACA0012 profile: Close up density contours in the stagnation area for a) FPM without weighted balancing diffusion , b) FPM with weighted balancing diffusion, c) RK solution and d) TG solution.

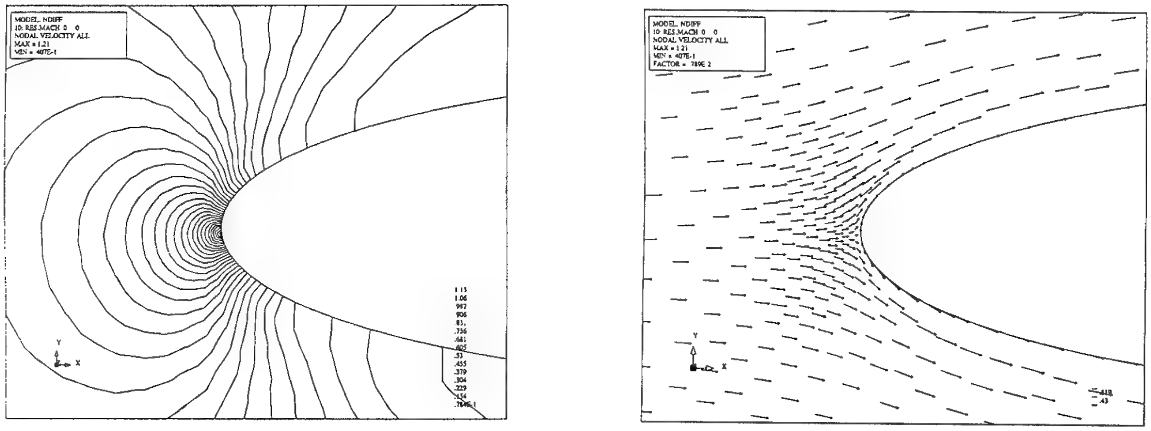


Figura 14: NACA0012 profile: a) velocity contours and b) velocity vectors in the stagnation zone are shown for the FPM with Gaussian weighted balancing diffusion terms.

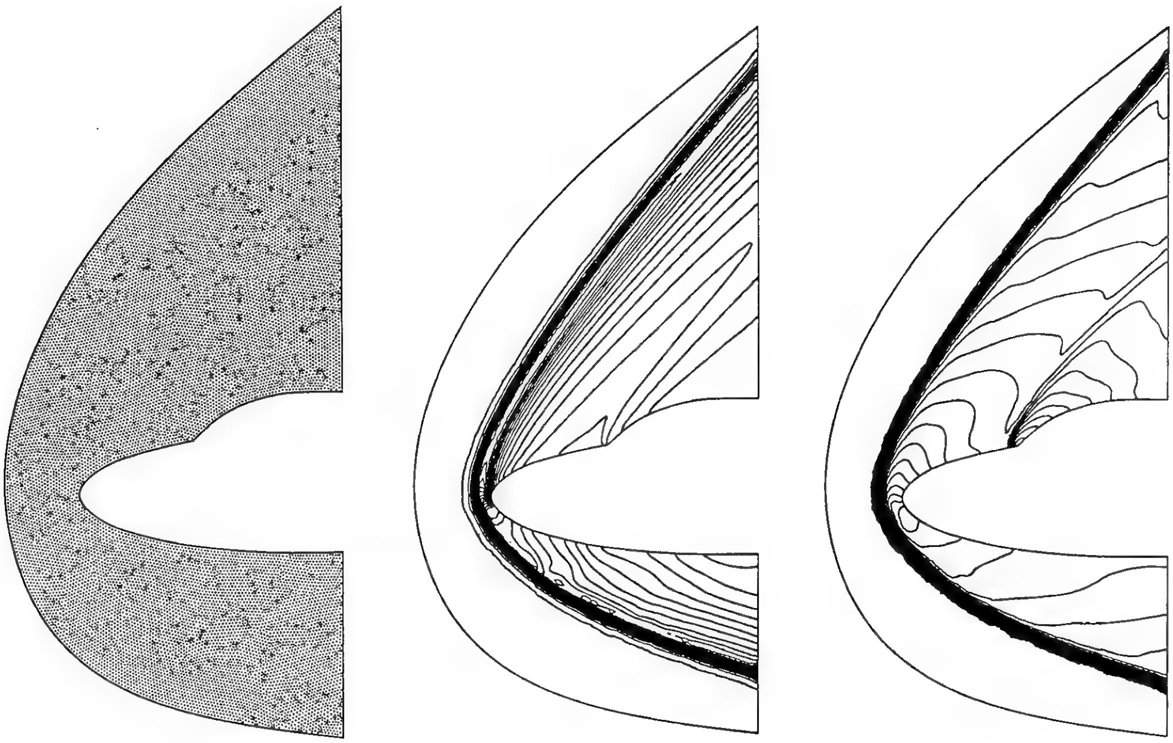


Figure 15: Double ellipse: a) Meshless grid, b)Mach number contours and c) Density contours for a hypersonic inviscid flow around a double ellipse.

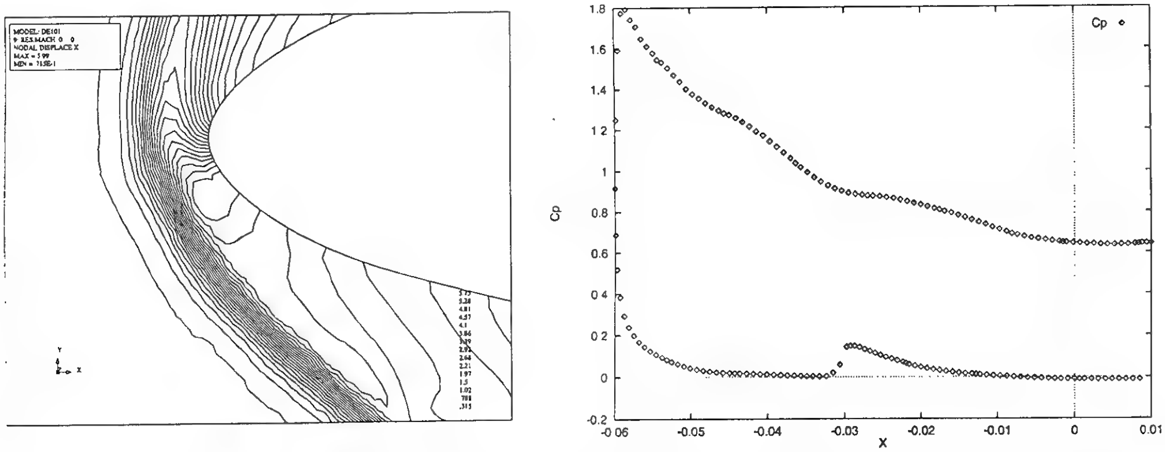


Figure 16: Double ellipse: a) Density contours in the stagnation zone and b) Pressure coefficient c_p along the boundary of the body.

NUMERICAL SIMULATION OF INTERNAL AND EXTERNAL GAS DYNAMIC FLOWS ON STRUCTURED AND UNSTRUCTURED ADAPTIVE GRIDS

U.G.Pirumov,
I.E.Ivanov,
I.A.Kryukov
Moscow Aviation Institute
Volokolamskoe sh. 4
125871 Moscow, Russia

1. ABSTRACT

Solution algorithms for solving the unsteady 2D Euler equations are presented. Cell-centered upwind control volume scheme are developed with utilize the two-dimensional monotone linear reconstruction procedures. The new adaptive grid procedure are proposed to cluster the grid points in regions where they are most needed. This procedure is generalized for unstructured grids. Numerical results in two-dimensional case are presented for linear and nonlinear convection problems.

2. INTRODUCTION

The numerical simulation of many gas dynamics processes, possessing applied significance, requires the solution unsteady two-dimensional Euler equations in the complex geometry region. The typical feature of inviscid gas flow about bodies, in channel of the complex form or in jets is presence interacted shock waves and other gas dynamics discontinuities [1-3]. For computation of such flows the high order schemes of TVD or ENO type are obtained wide spreading. These schemes have high order of accuracy in the region of the smooth solution, well capture discontinuities and preserve monotonicity of the solution. In the present paper for the solution of Euler equations high order version of Godunov's scheme [4,5] is used.

To improve efficiency of codes based on TVD and ENO methods and to resolve local features of a flow the solution-adaptive grid algorithms can be used. The authors have developed an adaptive grid algorithm suitable for structured and unstructured grids. It is based on the algebraic minimal moments scheme by Connett, etal. [6,7] with cell-centered grid modifications.

The proposed method belongs to the class of moving grid methods. Using these methods for structured grids, strongly skew cells can be obtained near large gradient regions. In this case, 1D- procedure along gridlines can yield large error due to decreasing of order of approximation. So it is necessary to use essentially 2D reconstruction procedures. Note that only 2D reconstruction procedures on unstructured grids can be used.

In present paper the linear reconstruction procedures are considered and one such procedure is developed. It is based on the well-known in Russia 2D algorithm by Tillyaeva [8] with modification which taking into account a more wide additional support (the set of cells needed to determine the coefficients of the polynomial).

Numerical results are presented in Section 4 to illustrate the capability of the proposed algorithms.

3. GOVERNING EQUATIONS

The governing equations are the conservation form of the Euler equations for two-dimensional, unsteady, compressible flows of a calorically perfect gas

$$q_t + F(q)_x + G(q)_y = S \tag{1}$$

where

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \; F(q) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{bmatrix}, \; G(q) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \end{bmatrix} \tag{2}$$

Here ρ, p and E are the density, pressure and total energy, respectively, and u and v are the Cartesian components of the velocity vector. S is the source term. The system (1)-(2) of four equations is closed with the polytropic equation of state

$$p = (\gamma - 1)(E - \rho / 2(u^2 + v^2)), \tag{3}$$

where γ is the ratio of specific heats.

2. CLASS OF HIGH ORDER SCHEME FOR NUMERICAL SIMULATION GAS DYNAMIC FLOWS

2.1 Finite volume formulation

Let denote by S_{ij} a rectangular partition of the x-y plane, where

S_{ij}=$[x_{i-1/2}, x_{i+1/2}]\times[y_{j-1/2}, y_{j+1/2}]$,

with (x_i, y_j) denoting centroid of each rectangle S_{ij}. With help of integral form of equation (1) for each control volume S_{ij} following equation can be obtained

$$\frac{\partial \bar{q}_{ij}(t)}{\partial t} = -\frac{1}{\Delta s_{ij}}\Big[\hat{f}_{i+1/2,j}(t) - \hat{f}_{i-1/2,j}(t) + \hat{g}_{i,j+1/2}(t) - \hat{g}_{i,j-1/2}(t)\Big], \tag{4}$$

where $\Delta s_{ij} = \Delta x_i \Delta y_j$ is area of S_{ij}, and

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

$$\bar{q}_{ij}(t) = \frac{1}{\Delta s_{ij}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} q(x,y,t)dxdy \tag{5}$$

is the cell average of q over the control volume at time t. The fluxes $\hat{f}$ and $\hat{g}$ are given by

$$f_{i+1/2,j}(t) = \int_{y_{j-1/2}}^{y_{j+1/2}} f(q(x_{i+1/2},y,t))dy, \tag{6}$$

$$g_{i+1/2,j}(t) = \int_{x_{i-1/2}}^{x_{i+1/2}} g(q(x,y_{j+1/2},t))dx. \tag{7}$$

Equation (4) can be treated as a system of ordinary differential equations. Along any t=constant line, the right-hand side of (4) is a spatial operation in q, and we rewrite this equation in the abstract operator form

$$\frac{\partial}{\partial t}\bar{q}_{ij}(t) = (\bar{L}q(t))_{ij} \tag{8}$$

for the purpose of "separation" the spatial and temporal discretizations.

2.1.1 Spatial discretization

To achieve desired order of accuracy we replace the operator $\bar{L}$ with a discrete spatial operator L, which approximates $\bar{L}$ to r-1 order

$$L\bar{q}(t) = \bar{L}q(t) + O(h^r). \tag{9}$$

We define L explicitly by

$$(L\bar{q}(t))_{ij} = -\frac{1}{a_{ij}}\Big[\bar{f}_{i+1/2,j}(t) - \bar{f}_{i-1/2,j}(t) + \bar{g}_{i,j+1/2}(t) - \bar{g}_{i,j-1/2}(t)\Big], \tag{10}$$

where f and g are approximations of corresponding order to fluxes $\hat{f}$ and $\hat{g}$ in (6) and (7).

For approximation of integrals (6), (7) we can use "classical" K-point Gaussian quadrature. Therefore, for fixed x and t, and sufficiently smooth f, the approximation of the flux integral (6) by Gaussian quadrature satisfies

$$\hat{f}_{i+1/2,j}(t) = \frac{\Delta y_j}{2}\sum_{k=1}^{K} C_k f(q(x_{i+1/2},y_k,t)) + s(x_{i+1/2},\eta)h^{2K+1}, \tag{11}$$

where function s relates to the quadrature error and $\eta \in (y_{j-1/2}, y_{j+1/2})$.

Let R be a spatial operator which reconstructs the set of cell average and yields a 2D, piecewise polynomial $q_h(x,y)$ of degree r-1 which approximates $q(x,y,t)$, with a truncation error of $O(h^r)$

$$q_h(x,y) = R(x,y,\bar{q}(t)). \tag{12}$$

Therefore, if we define "abstract" numerical flux f in (10) by

$$\bar{f}_{i+1/2,j}(t) = \frac{\Delta y_j}{2}\sum_{k=1}^{K} C_k f(q_h(x_{i+1/2},y_k)), \tag{13}$$

Analysis (11),(13) and the flux difference $f_{i+1/2,j}(t) - f_{i-1/2,j}(t)$ in (10) shows [9], that number of Gauss points K must satisfied condition $r \leq 2K$. Then the error relation satisfies

$$\hat{f}_{i+1/2,j}(t) - \hat{f}_{i-1/2,j}(t) = \bar{f}_{i+1/2,j}(t) - \bar{f}_{i-1/2,j}(t) + O(h^{r+2}). \tag{14}$$

Noting that the area Δs_{ij} is $O(h^2)$, than upon substitution of the numerical fluxes (13) into (10) we have thus designed the spatial operator L that satisfied (9).

We now wish to modify the "abstract" numerical flux (13) such that conditions of approximation of desired order for scheme be satisfied in regions where the solution is smooth and, in addition, these fluxes will account for possible discontinuities in q. This modification follows naturally from the reconstruction procedure, by which the function $q_h(x,y)$, in (12), can discontinuous at cell interfaces. presentation of $q_h(x,y)$ within a cell S_{ij}. In order to resolve these discontinuities, the flux integrands in (13) are replaced by

$$f^{RP}(\tilde{q}_{ij}(x_{i+1/2},y_k), \tilde{q}_{i+1,j}(x_{i+1/2},y_k)),$$
$$g^{RP}(\tilde{q}_{ij}(x_k,y_{j+1/2}), \tilde{q}_{i,j+1}(x_k,y_{j+1/2})), \tag{15}$$

where $\tilde{q}_{ij}(x,y)$ denote the local representation of $q_h(x,y)$ within a cell S_{ij} and $f^{RP}(q_1,q_2)$ denotes the flux, across $x=0$, associated with the solution to the Riemann problem whose initial states are q_1 and q_2.

2.1.2 Temporal discretization

Equation (4) is discretized by using a Runge-Kutta method (R-K) of Shu [10]:

$$\bar{q}_{ij}^{(l)} = \sum_{m=0}^{l-1}[\alpha_{lm}\bar{q}_{ij}^{(m)} + \beta_{lm}\Delta t\, L_{ij}^{(m)}], \quad l=1,2,\ldots,p,$$
$$L_{ij}^{(m)} = (L\bar{q})_{ij}^{(m)}, \quad \bar{q}_{ij}^{(0)} = \bar{q}_{ij}^{(n)}, \quad \bar{q}_{ij}^{(p)} = \bar{q}_{ij}^{(n+1)}. \tag{16}$$

The order of accuracy, as well as its TVD properties, is achieved by adequate sets of coefficients α_{lm}, β_{lm} and p [10].

Also in our code method [11] is used. It is predictor-corrector type method of second order accuracy, in which on first stage the fluxes are calculated without solution of the Riemann problem and reconstruction procedure is performed only on first stage.

2.1.3 Riemann solvers

Using of the solution of the Riemann Problem (RP) for calculation of the fluxes over faces of a control volume (16) allows to take into account local directions of perturbation propagation. In general case it is necessary to use really two dimensional Riemann Solver for accurate calculation of 2D flows. But recently proposed 2D Riemann Solvers [12,13] are too complicated, tedious and sometime result in instability, even for the first order schemes. So in presented paper the solution of RP is determined by 1D Riemann Solvers. According to common practice, which start from works of S.K. Godunov [5], the normal to face of a control volume is used as the direction, along which the 1D RP is solved. It can result in dependence of the solution quality on numerical grid.

In developed computer code [14] there is opportunity to choice of the 1D Riemann Solver (for example exact solution [4] or approximate methods by Roe [15], Osher [16], Dukowicz [17], Davis [18] and others [19]) for the solution of the RP.

2.1.4. Boundary conditions

Correct setting of numerical boundary conditions is one of the most important items for the numerical simulation of unsteady gas dynamics flows. Numerical boundary conditions based on using of characteristic relations are most correct in physical sense and they are implemented in the present work. For the open boundaries the "non-reflecting" boundary conditions [20] are used.

2.2. Two dimensional reconstruction algorithms

Using solution-adaptive grid methods for structured grids, strongly skew cells can be obtained near large gradient regions. In this case, 1D- procedure along gridlines can yield large error due to decreasing of order of approximation. So it is necessary to use essentially 2D reconstruction procedures [8,9,21,22,25,26]. Most of these procedures are rather complicated and very costly. At present paper most simple linear reconstruction procedures are used [8,22]. They can efficiently realized on the structured grids.

In the reconstruction procedure by Tillyaeva [8] (denoting below as TL) the five-point stencil is decomposed into triangles. The slopes of planes passing through function values at the vertices of each of the four triangles with a vertex at *i,j* are determined. The derivatives with respect to x and y of linear reconstructed function are evaluated from corresponding slopes by **minmod** operator. As shown in [8] for case of structured quadrilateral grids only two opposite triangles with a vertex at *i,j* can be used in the reconstruction procedure. Moreover, on rectangular grids the reconstruction became couple of the independed 1D **minmod** limiters.

As follow from numerical results TL reconstruction is too much diffusive. So we modify it by taking into account slopes of central plane, reconstructed with algorithm similar [21] over all points of the stencil. We denote this algorithm with MT.

Last algorithm that studied in this paper is the linear reconstruction proposed in [22]. Initially the estimate of solution gradient in the cell are computed using the approximation of boundary integral for some path surrounding this cell. Then the obtained gradient estimation is restricted to satisfy the monotonicity principle.

3. ADAPTATION PROCEDURE

The adaptive grid algorithm presented here is based on the algebraic minimal moment method [6,7] with cell-centered grid modifications. For exact reconstruction of a linear function the centroid of control volume, where averaged function values are stored, has to coincide with the gravity center of this control volume. So in presented algorithm firstly vertices of control volumes are replaced and then coordinates of gravity centers of new control volumes are evaluated.

An analytical expression for the movement of cell vertex *P* is given as

$$\Delta\vec{P} = \vec{P}_{new} - \vec{P} = \frac{1}{M}\sum_{nb} m_{nb}(\vec{B}_{nb} - \vec{P}), \tag{17}$$

$\sum$ - means summation over all neighbour cells with a vertex at $\vec{P}$. $\vec{B}$ is the vector location of the gravity center of corresponding cell.

The "mass" of cell *n* is defined as

$$m_n = \max_j\left\{\frac{|f_n - f_j|}{\|B_n - B_j\|} + c\right\}\Delta s_n, \tag{18}$$

where *j* is index of adjusting with *n* cells, Δs_n is the cell area and

$$M = \sum_{nb} m_{nb}. \tag{19}$$

The first term within the parenthesis of equation (18) represents an estimate of the maximum gradient of an arbitrary function, *f*, on a grid cell. The adaptation will be sensitive to the gradient of this function. The constant *c* is the user specified constant which controls the adaptation strength.

For boundary point adjustment two algorithms proposed in [6,23] are used.

In the method of minimal moments [6,7] the nine-point stencil is decomposed into triangles. The gravity center for each of the four triangles with a vertex at *i,j* is determined. A weighting function value is computed for these points. The new location of point *i,j* is at common center of mass of these four triangles and is given by (18) with summation over these triangles. In the presented method "natural" information is mainly used: gravity centers and areas of computation cells, that are computed and stored for a flow field computation. In some cases an estimate of the maximum gradient in the weighting function (18) can be obtained as auxiliary result of a reconstruction procedure. It can be rather important for overall efficiency of flow solver with dynamic solution-adaptive grid techniques.

The presented method can be easy generalized on unstructured cell centered grids. The illustrations in Fig. 1 were produced to show how presented algorithm can be used on structured quadrilateral and unstructured triangular grids. Fig. 1a presents surface plot of given function H(x,y)=tanh(3(x-y^2))

+tanh(3(x2+y2-1)). Figures 1b and 1c contain the solution-adaptive grids produced by our algorithm.

4. NUMERICAL RESULTS

4.1. Linear scalar 2D problems

Numerical dissipation of explicit high resolution schemes with various "reconstruction" algorithm were compared by numerical results for the two dimensional advection equation [24]

$$u_t + (\alpha(y)u)_x + (\beta(x)u)_y = 0 \qquad (20)$$

$$\text{with} \quad \alpha(y) = -(y - y_0)\omega, \ \beta(y) = -(x - x_0)\omega \qquad (20)$$

The exact solution of (20), (20) consist in the rotation of the initial values round (x_0, y_0) with angular velocity ω. In this paper presented two series of calculations. As initial values was chosen a cut-out cylinder and a cone (fig. 2). We used the angular velocity ω to be 0.1 and x_0=50, y_0=50. The region of computation was [0,100]×[0,100]. The numerical calculation were done on three type of grids with 100 grid points in each direction. The first type is uniform rectangular grid, the second type is smooth curvilinear grid (Fig. 3.a) described by transformation

$$x_{ij} = \xi_i - \frac{50}{4\pi}\sin(\frac{\pi}{50}(\xi_i - \eta_j)),$$

$$y_{ij} = \eta_j - \frac{50}{4\pi}\sin(\frac{\pi}{50}(\eta_j - \xi_i)),$$

where $\xi_i = \Delta x(i-1), \quad \Delta x = 1,$

$\eta_j = \Delta y(j-1), \quad \Delta y = 1.$

And the third type is random grid (Fig. 3.b)

$$x_{ij} = \xi_i + \varepsilon_{ij}\Delta x$$

$$y_{ij} = \eta_j + \omega_{ij}\Delta y$$

where ε_{ij} and ω_{ij} are uniformly distributed random numbers on (-0.4, 0.4).

At time t=20π the initial values have carried out one full rotation and returned to their initial position. The approximations of the initial values on uniform grid are shown in Fig. 2. To improve picture resolution in Figs. 2 and 4 we used only part of the computation region [50,100]×[25,75]. Size and initial position of the cut-out cylinder and the cone are same as in [24]. We perform long time calculations until t=120π which corresponds to six full rotations of initial values. As mentioned in [24] these problems are well suited to benchmark the numerical properties of the schemes.

In this paper three reconstruction procedures described in section 2.2 are compared. Numerical results obtained with these reconstructions on computational grids of three types are presented in Tab. 1 for the cone and in Tab. 2 for the cut-out cylinder. These tables contain numerical solution errors calculated with respect to the L_1 norm and maximum values of obtained solutions. Fig. 4 shows numerical solutions computed using uniform rectangular grid. TL reconstruction results are shown in Figs. 4a and 4d, MT results - Figs. 4b and 4e and BJ results - Figs. 4c and 4f. As we expect the TL reconstruction procedure is most dissipative. MT and BJ results are rather close, but in MT case errors are a little smaller and maximum values are a little greater. These problems also shows that the MT reconstruction may not preserve the symmetry. We have some difficulties with implementation of the BJ reconstruction on the curvilinear grids. So results for this grid aren't presented in Tabs. 1 and 2.

4.2. A channel with a 15° compression-expansion ramp

The next case is the flow through a duct with compression-expansion ramp in the bottom wall. The conditions for this case are: M_∞=2, γ=1.4. The computational grid is equally spaced and contains 180 cells in the streamwise direction and 60 cells in the cross flow direction.

The computed Mach contours are presented in Fig. 4. Note that the induced and reflected shocks are quite thin and any unphysical oscillations are absent. All of characteristic features of the flow are well resolved on such fine grid without adaptation.

4.3. A Oblique Shock-reflection problem

One of the most popular problems for checking out various elements of numerical algorithm (such as reconstruction, adaptation etc.) is the regular reflection of an oblique shock wave by a flat plate. In Figs. 5 a-f, results are shown for a case with M_∞=2.9 and $\beta = 29^o$, where β is the angle made by incident shock wave and the flat plate Fig. 5a. First steady solution was obtained on the uniform 60×20 rectangular grid. The corresponding pressure contours are presented in Fig. 5a. Then grid adaptation was performed by the proposed above method. The pressure was used as the adaptation function. After that new steady state was obtained. Figs. 5a and 5b depict the adapted grid and associated steady state flow solution. These figures shows that the pressure gradients become much better resolved.

4.4. A underexpanded jet flow

The next case is the unsteady underexpanded supersonic jet flow. The conditions for this case are: M_j=1.5, n=p_j/p_∞=3, T_j=T_∞ and γ_j=γ_∞=1.4. The computational grid is equally spaced and contains 180 points in the streamwise direction and 80 points in the crossflow direction. Fig. 6 shows the computed Mach contours for most characteristic time moments. Note that the nonreflecting boundary conditions allow to calculate such rather complicated flow almost without unphysical reflection on the open boundaries.

The steady state solution is shown in Fig. 7a. Fig. 7b shows the steady state solution obtained using coarse 90×40 rectangular grid. For this solution the grid adaptation was performed using the Mach number gradients as the adaptation function. In Fig. 7c, the adaptive grid is presented and Fig. 7b shows the computed Mach contours. The solution computed using coarse adapted grid is mush close to the fine grid solution in Fig. 7a. But some flow features aren't succeeded to capture. The second Mach stem places somewhat farther from the nozzle cut than in Fig. 7a.

But the first Mach stem are resolved better than by using the fine grid. It is main deficiency of moving solution-adaptive grid algorithms. If there are large gradients of parameters in a flow region then grid points are too coarse in regions of middle and low gradients.

4.5. Nozzle flow

In the last example we present results of numerical simulation of the internal axisymmetric nozzle flow of the ideal gas with γ=1.22. The nozzle consists from two parts. First part is a Laval nozzle and second one is cylindrical tube adjoining to the supersonic part of the Laval nozzle.

Initially the steady state solution was obtained using 130×40 simple grid (Fig. 8a). In Figs. 8a and 8b the top half of figures shows the computed Mach contours and the bottom half shows the computational grid. For the computed steady state solution the grid adaptation was performed using the Mach number gradients as the adaptation function. The new steady state solution and the adaptive grid are presented in Fig. 8b.

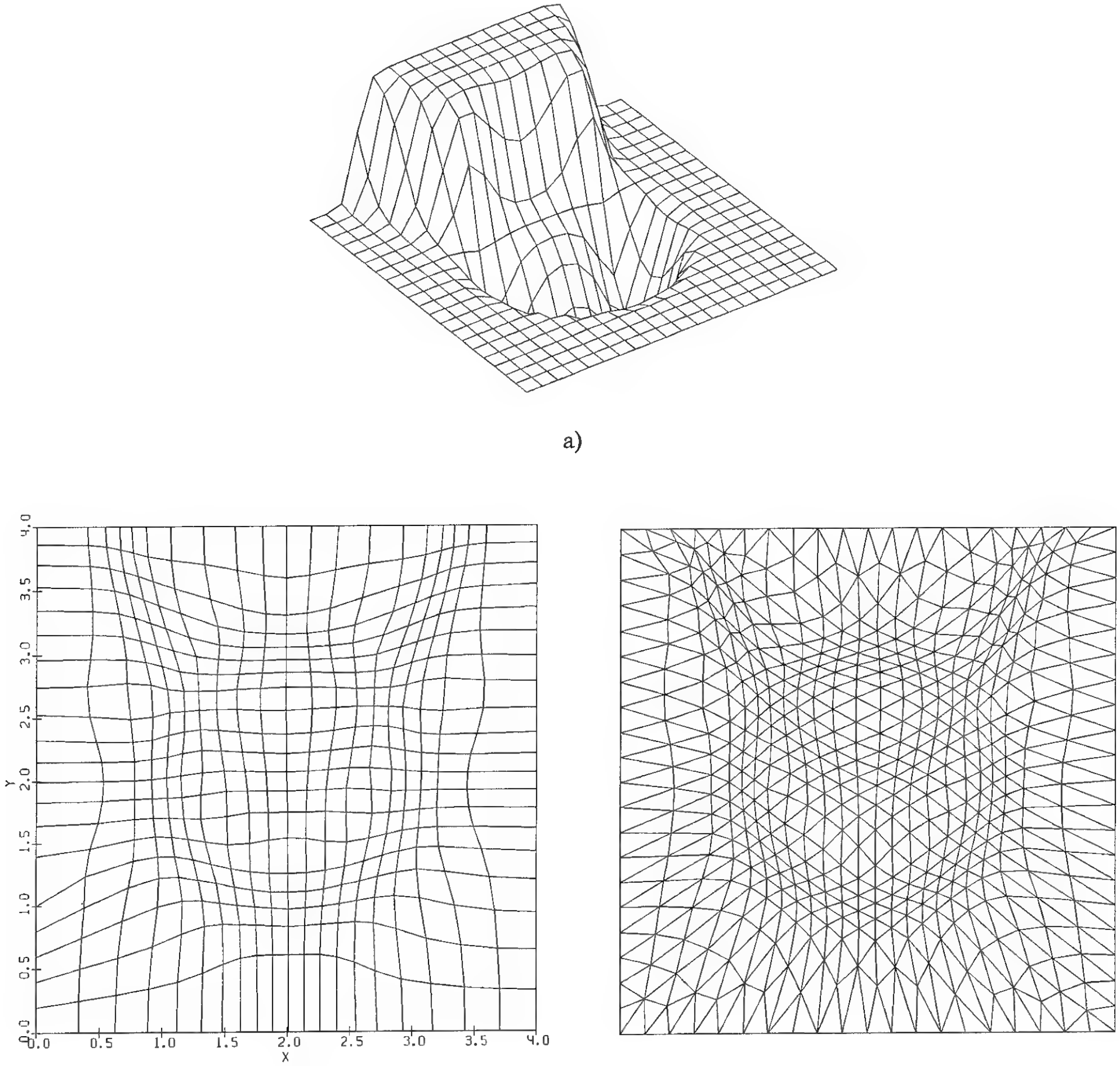
5. CONCLUSIONS

In present paper the upwind monotone numerical method for the solution of the Euler equations is presented. This method is based on the high order version of the Godunov's scheme. This method is realized using both the structured quadrilateral and the unstructured triangular grids. Essentially 2D reconstruction procedures make possible to perform calculations using the strongly skew grids. Some features of 2D reconstruction procedures are studied to solve the linear scalar problem. The new solution-adaptive grid algorithm is proposed.

Presented numerical results illustrates the capability of the proposed algorithms. It can be see that the grid adaptation procedure make possible to obtain significantly more accurate results.

6. REFERENCES

1. Pirumov U.G., Roslyakov G.S., "Gas Flows in Nozzles", Springer-Verlag, 1985, 425 p.
2. Gorbunov V.N., Pirumov U.G., Ryzhov Yu.A., "Non-Equilibrium Condensation in High-Speed Gas Flows", Gordon and Breach Science Publishers, 1989, 290 p.
3. Pirumov U.G., Roslyakov G.S., "Numerical Method for Gas Dynamic", Moscow, Vishai Shkola, 1987, 231 p., (in Russian).
4. Godunov S.K., "Finite Difference Method for Numerical Calculation the Discontinue Solutions of Gasdynamic Equations", Mathem. Sb., 47, 3, 1959, pp. 271-306, (in Russian).
5. Godunov S.K., Zabrodin A.V., Ivanov M.Ya., Krayko A.N., Prokopov G.P., "Numerical Solution of Multidimensional Gasdynamics Problems", Moscow, Nauka, 1976, 400 pp., (in Russian).
6. Connett W.C., Agarwal R.K., Schwartz A.L., "An Adaptive Grid-Generation Scheme for Flowfield Calculations", AIAA Pap., 87-0199, 1987.
7. Connett W.C., Agarwal R.K., Schwartz A.L., Wheeler J.C., "An Algebraic Adaptive Grid Technique for the Solution of Navier-Stokes Equation", AIAA Pap., 90-1605, 1990.
8. Tillyaeva N.I., "Generalization of Modified Godunov's Scheme for Unstructured Grids", Uchenyie Zapiscki TsAGI, XVII, 2, 1986, pp. 18-26, (in Russian).
9. Casper J., Atkins H.L., "A Finite-Volume High-Order ENO Scheme for Two-Dimensional Hyperbolic Systems", J.Comput.Phys., 106, 1993, pp. 62-76
10. Shu C-W., Osher S. "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes", J.Comput.Phys., 77, 1988, pp. 439-471.
11. Rodionov A.V., "High Order Godunov Scheme", Zhurnal vichislitelnoy mathem. i mathem. phys., 27, 4, 1987, pp. 585-593, (in Russian).
12. Ramsey C.L., van Leer B., Roe P.L., "A Multidimensinal Flux Function with Application to the Euler and Navier-Stokes Equation", J. Comput. Phys., 105, 1993, pp. 306-323.
13. Roe P.L., "Discrete Model for the Numerical Analysis of Time-Dependent Multidimensional Gas Dynamics", J.Comput.Phys., 63, 1986, pp. 458-476.
14. Kryukov I.A., Ivanov I.E., "High Resolution Monotone Method for Computation Internal and Jet Inviscid Flows", in "Nonequilibrium processes in nozzles and jets", Proc. 1st Int. Conf., Coll. Abstr., June 1995, pp. 92-93.
15. Roe P.L., "Approximate Riemann Solver Parameter Vector and Difference Schemes", J.Comput.Phys., 43, 1981, pp. 357-372.
16. Osher S., Solomon F., "Upwind Difference Schemes for Hyperbolic Systems of Conservation Laws", Math.Comput., 38, 158, 1982, pp. 339-374.
17. Dukowicz J.K.,"A General, Non-iterative Riemann Solver for Godunov's Method", J.Comput.Phys., 61, 1985, pp. 119-137.
18. Davis S.F., SIAM J.Sci.Stat.Comput., 9, 1988, pp. 445-473.
19. Einfeldt B., SIAM J.Numer.Anal., 25, 1988, pp. 294-318.
20. Thompson K.W., "Time Dependent Boundary Conditions for Hyperbolic Systems", J.Comput.Phys., 68, 1, 1987, pp. 1-24.
21. Vankeirsbilck P., Deconinck H., "Solution of the Compressible Euler Equations with Higher Order ENO-schemes on General Unstructured Meshes", in "Computational Fluid Dynamics'92", vol.2, 1992.
22. Barth T.J., Jespersen D.C., "The Design and Application of Upwind Schemes on Unstructured Meshes", AIAA Pap, 89-0366, 1989.
23. Kania L.A. "An Adaptive Grid Algorithm for Accurate Flowfield Calculations", AIAA Pap., 90-0327, 1990.
24. Munz C-D., "On the Numerical Dissipation of High Resolution Scheme for Hyperbolic Conservation Laws", J.Comput.Phys., 77, 1988, pp. 18-39.
25. Harten A., Chakravarthy S.R., "Multidimensional ENO Schemes for General Geometries", Tech., Report 91-76, ICASE, 1991.
26. Abgrall R. "On Essentially Non-oscillatory Schemes on Unstructured Meshes: Analysis and Implementation", J.Comput.Phys., 114, 1994, 45-58.



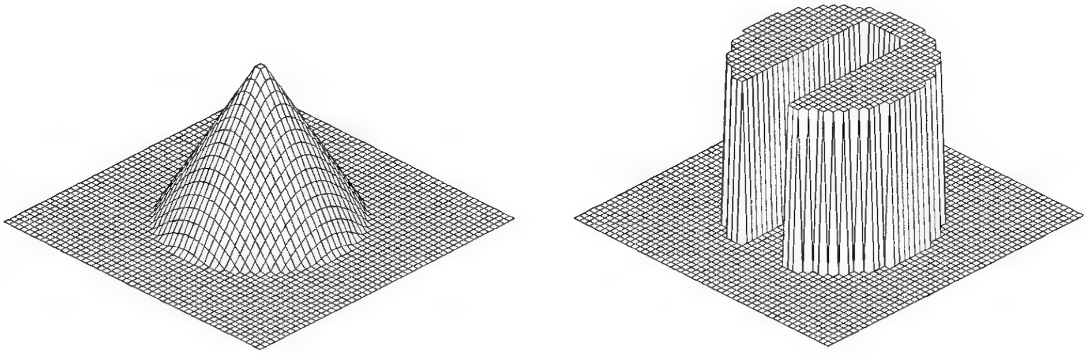
a)

b)

c)

Fig. 1. Adaptive response of uniform grids to weighting function H(x,y)=tanh(3(x-y^2)) + tanh(3(x^2+y^2-1)).

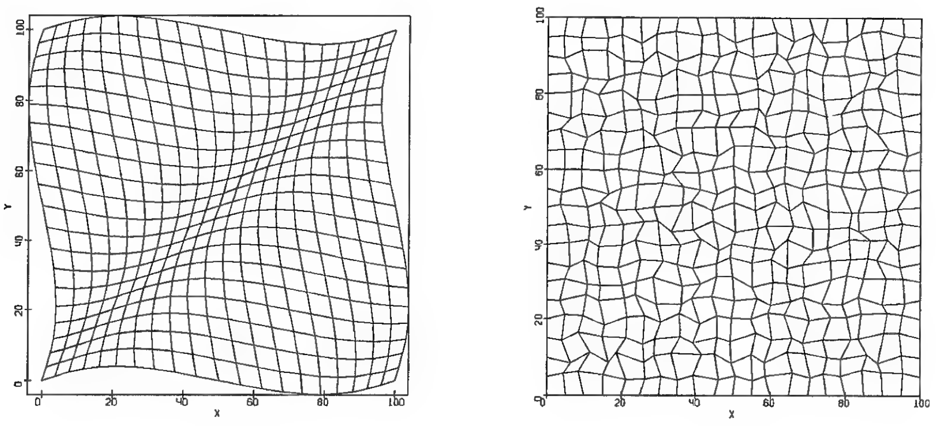
a) weighting function; b) regular quadrilateral grid; c) unstructured triangular grid.



a)

b)

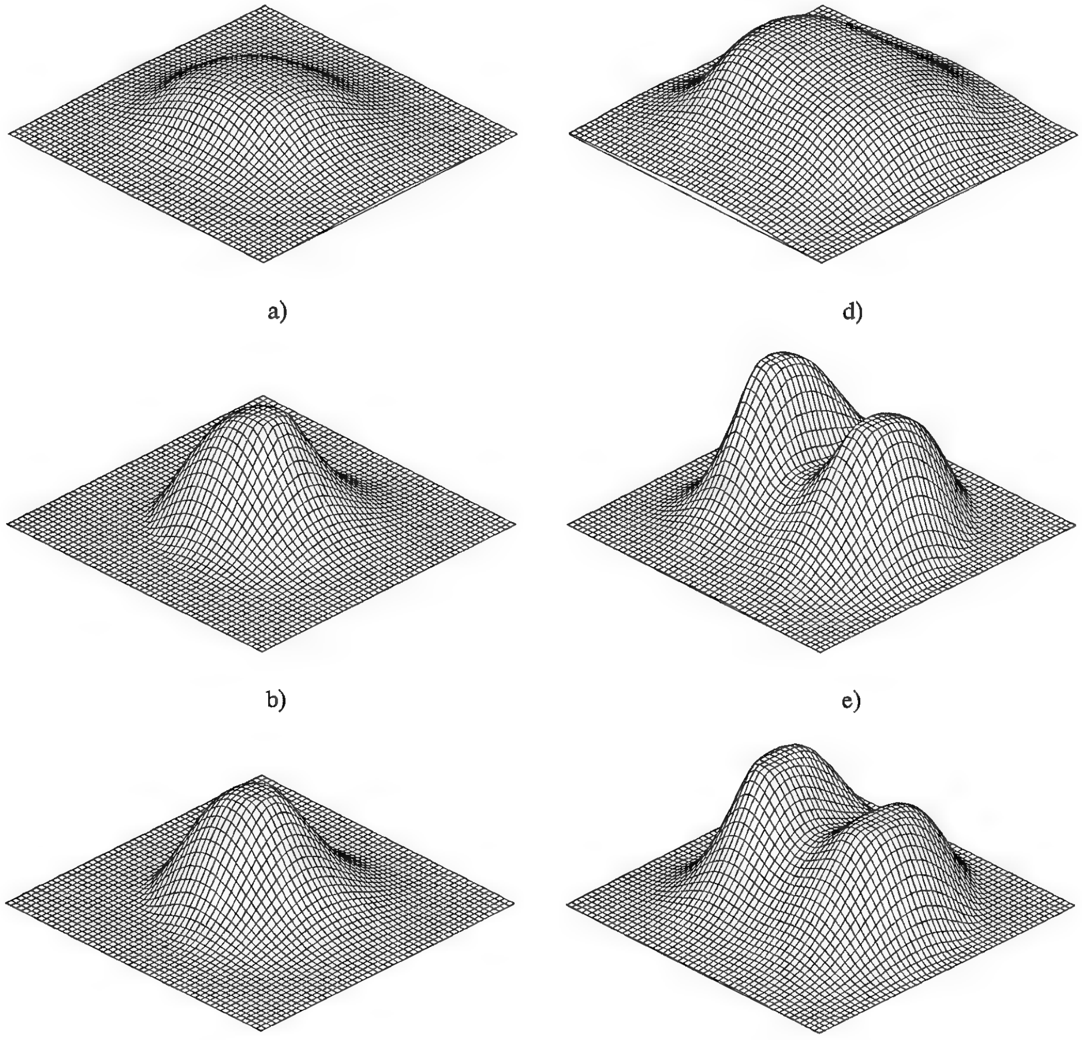
Fig. 2. Initial values and exact solution after each full rotation. a) - cone, b) - cut-out cylinder.



a)

b)

Fig. 3. Examples of curvilinear (a) and random (b) calculation grids (20×20).



a)

d)

b)

e)

c)

f)

Fig. 4. Computed solutions on the uniform grid after six rotations. a)-c) - cone, d)-f) - cut-off cylinder.

	Type of grid					
Reconstruction	Uniform grid		Curvilinear grid		Random grid	
	L_1 error	Max value	L_1 error	Max value	L_1 error	Max value
TL	0.0616481	1.48475	0.0675634	1.22261	0.0640970	1.43263
MT	0.0216717	2.80456	0.0245567	2.45170	0.0257393	2.85844
BJ	0.0246873	2.74293	-	-	0.0261619	2.65865

Table 1. Results of solution of the linear scalar problem for rotating cone.

	Type of grid					
Reconstruction	Uniform grid		Curvilinear grid		Random grid	
	L_1 error	Max value	L_1 error	Max value	L_1 error	Max value
TL	0.198722	2.10699	0.188287	2.05245	0.195201	2.02538
MT	0.206402	3.66130	0.201828	3.39611	0.208682	3.58780
BJ	0.206362	3.18942	-	-	0.203615	3.04373

Table 2. Results of solution of the linear scalar problem for rotating cut-out cylinder.

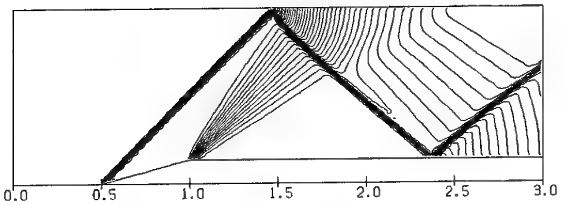
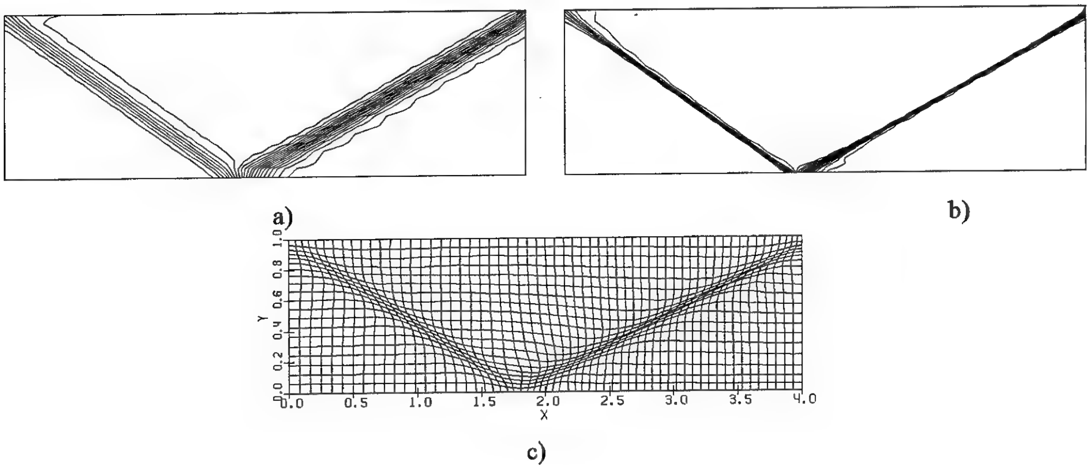


Fig. 5. Computed pressure contours for a 15° compression ramp (M=2).



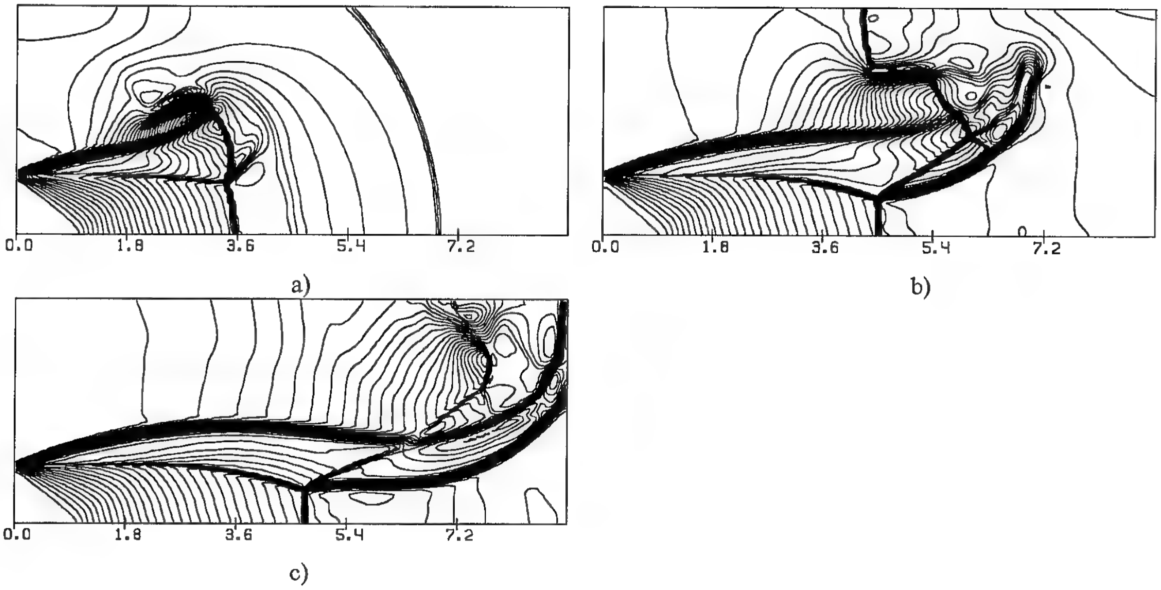
a)

b)

c)

Fig. 6. Pressure contours for oblique shock reflection problem (M=2.9, β=29°).

a) solution on uniform grid; b) solution on adaptive grid; c) solution-adaptive grid.

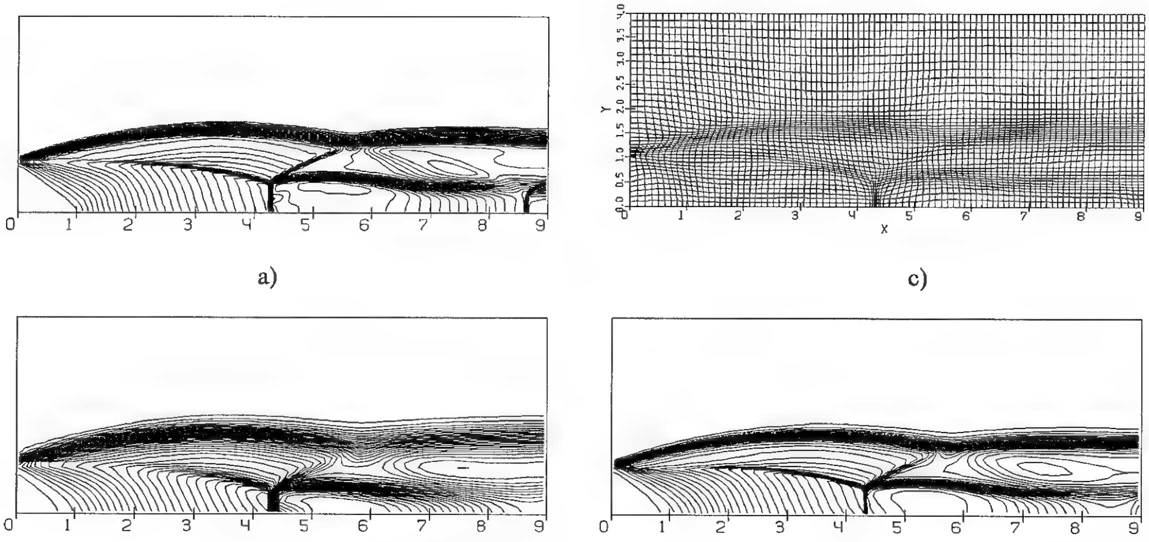


a)

b)

c)

Fig. 7. Unsteady underexpanded (n=3) jet.



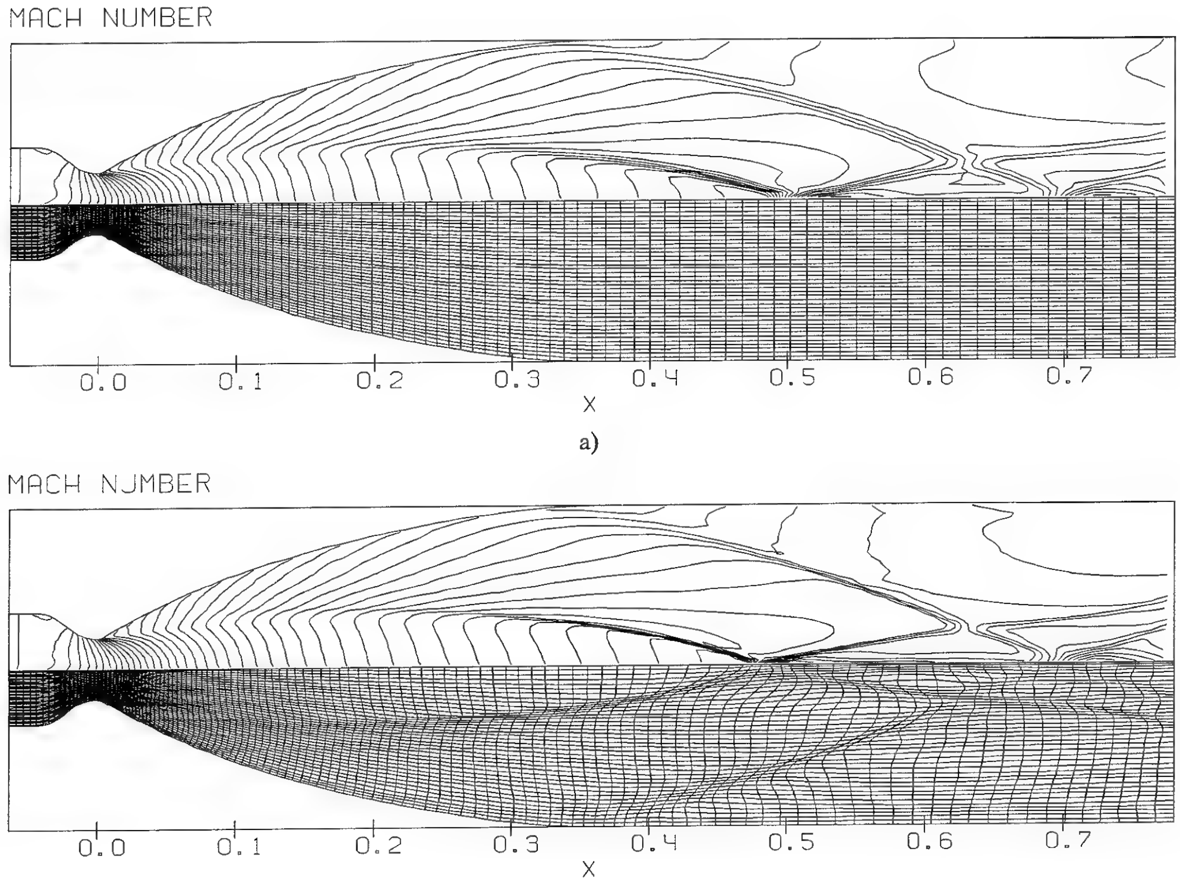
a)

c)

b)

d)

Fig. 8. Steady underexpanded jet. a) - fine grid solution; b) - coarse grid solution without adaptation; c) - adaptive grid; d) - adaptive coarse grid solution.



a)

b)

Fig. 9. Nozzle flow problem. a) solution and grid without adaptation; b) adaptive solution and grid.

An Investigation of The Effects of the Artificial Dissipation Terms in a Modern TVD scheme on the Solution of a Viscous Flow Problem.

R. D. Briggs[1] and S.Shahpar[2].
The Manchester School of Engineering,
Aerospace Engineering Division,
Oxford Road, Manchester. M13 9PL
England

1. SUMMARY

The goal of the present investigation is to discover the effects of certain parameters in a modern TVD scheme on the solution of a viscous flow problem. This report includes details of the TVD scheme used in this study. The scheme is an extension of the work of H. C. Yee [1], and uses an upwind weighted dissipation term, and central differencing to calculate the viscous terms.

The entropy correction parameter and the choice of flux limiter when computing viscous flows are under investigation. The effectiveness of this TVD scheme in solving viscous flow problems has recently been questioned by Lin [2]. However, this investigation shows that by carefully selecting the limiter and the value of the entropy parameter, adequate viscous flow results can be obtained.

Solutions to the Navier-Stokes equations for an underexpanded sonic jet on a flat plate with a supersonic crossflow are used to illustrate the method. The test conditions were M_∞=2.61, and Re=749,000, and the boundary layer was considered to be laminar everywhere. The numerical code has been evaluated for this test case using experimental data presented by Zukoski and Spaid [3].

The study includes a qualitative analysis of the amount of artificial viscosity added by the TVD algorithm compared to the real viscosity; an investigation of the effects of the artificial viscosity term on the solution, including changes in pressure and skin-friction distribution along the surface of the flat plate, and the change in the boundary layer separation point for different values of the entropy parameter.

2. INTRODUCTION

Over the past decade a vast range of Total Variation Diminishing (TVD) schemes have become available, and have been widely used. These schemes use some method of intelligent switching to put artificial dissipation into a problem where it is needed or to conversely remove artificial dissipation from areas of a problem where it is not. In this way the shock capturing capabilities of Euler solving numerical schemes has improved enormously.

When the Navier-Stokes equations are being solved however, the interaction between the real viscosity and the artificial viscosity, provided by these modern schemes, must be considered and evaluated.

A great deal of work has been done on various aspects of the artificial dissipation added in a scheme, and how it affects the solution. Allmaras [4] has examined a boundary layer in a subsonic flow, and reports that upwind schemes using a matrix dissipation technique, are generally better than central difference schemes using a scalar dissipation formulation, for producing good boundary layer profiles. He also reports a slight velocity overshoot within the boundary layer, even using the upwind schemes. Tatsumi et al [5] have concluded that scalar switching schemes can produce good accuracy if a flux limiting technique is included. Tatsumi et al also suggests that anti-diffusive schemes will produce overshoots as described in [4] unless a flux limiter is included in the algorithm. Caughey and Varma [6] have used an integral technique to measure the effects of artificial dissipation on the flow calculations around a transonic aerofoil. They found that the dissipation errors and the total numerical errors were of a comparable magnitude, but in general the dissipation errors were larger than the total numerical errors. They also found that a Mach number scaling technique, for reducing the artificial dissipation being added, did reduce the dissipation related errors in most cases. Turkel and Vatsa [7] have compared a scalar artificial dissipation model with a matrix dissipation model on a 3D transonic aerofoil. They found that the matrix dissipation model improved the accuracy of the scheme. The accuracy was comparable to that produced by an upwind TVD scheme. It has also been reported that the matrix dissipation technique in a central difference scheme can produce high resolution results in a viscous flow, and that the matrix dissipation is essential for this behaviour, Swanson and Turkel [8].

In this work, a TVD artificial dissipation switching method which is similar to a matrix dissipation technique has been studied. This switch is controlled by two parameters, the choice of flux limiter, and the value of the entropy correction parameter, which has an indirect effect.

In order to study the effects of these terms we have considered a complex viscous flow problem, as further irregularities caused by artificial dissipation may present themselves in such a problem. An underexpanded jet interaction with a supersonic crossflow has been chosen. This problem includes a separated boundary layer and regions of recirculation, see figure 1. The effect of the artificial dissipation terms on these flow phenomena have not been examined before.

Jet injection into a crossflow is an important aerodynamic problem and has many uses in the aerospace industry. Reaction control systems on space vehicles and missiles are transverse jet problems. Jets for vortex control on the forebody are now being considered for high angle of attack control systems for fighter aircraft. Fuel injection in

[1] Postgraduate Student
[2] Post-doctoral research fellow, M AIAA

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

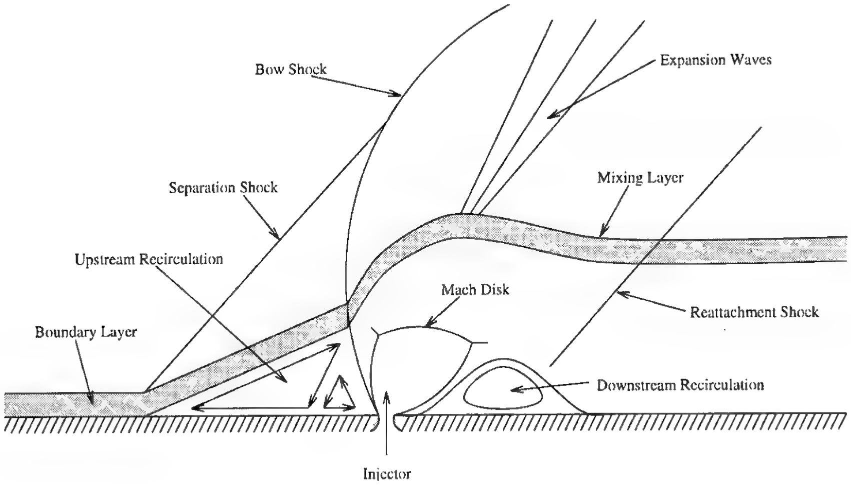


Figure 1. Diagram of a jet interaction with a supersonic crossflow.

supersonic ramjets (scramjets) can also be considered as a jet in supersonic crossflow problem.

The flowfield around a jet injection is shown in figure 1. An underexpanded jet transversely injected from the wall, into a supersonic main air flow, expands rapidly through the strong Prandtl-Meyer fans and forms a Mach disk. A bow shock wave is formed upstream of the injector due to the interaction between the main and injected flows while the main flow bends the injected flow parallel to the wall. A boundary layer separation occurs and a weak separation shock wave appears upstream near the injector due to an adverse pressure gradient caused by the injection in a boundary layer flow. Another weak shock appears downstream near the injector due to a reattachment of the injected flow with the wall.

The objectives of this present study were to measure the effect of the artificial dissipation switching algorithm on the accuracy of the solutions produced for this test case. Also, a quantitative and qualitative examination of the amount of real viscosity compared with the artificial viscosity being added in the solution has been included.

3. NUMERICAL METHOD

3.1 Governing Equations

For a Cartesian coordinate system the Navier-Stokes equations can be written in a conservative form:

$$\frac{\partial \hat{U}}{\partial t}+\frac{\partial \hat{F}}{\partial x}+\frac{\partial \hat{G}}{\partial y}=\frac{\partial \hat{F}_V}{\partial x}+\frac{\partial \hat{G}_V}{\partial y} \qquad (1)$$

where the inviscid terms are,

$$\hat{U}=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ e\end{pmatrix},\quad \hat{F}=\begin{pmatrix}\rho u\\ p+\rho u^2\\ \rho uv\\ (e+p)u\end{pmatrix},\quad \hat{G}=\begin{pmatrix}\rho v\\ \rho uv\\ p+\rho v^2\\ (e+p)v\end{pmatrix} \qquad (2)$$

and the viscous terms are,

$$\hat{F}_V=\begin{pmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ f\end{pmatrix},\qquad \hat{G}_V=\begin{pmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ g\end{pmatrix} \qquad (3)$$

where,

$$\tau_{xx}=\mu\left(\frac{4}{3}\frac{\partial u}{\partial x}-\frac{2}{3}\frac{\partial v}{\partial y}\right) \qquad (4a)$$

$$\tau_{xy}=\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) \qquad (4b)$$

$$\tau_{yy}=\mu\left(-\frac{2}{3}\frac{\partial u}{\partial x}+\frac{4}{3}\frac{\partial v}{\partial y}\right) \qquad (4c)$$

$$f=u\tau_{xx}+v\tau_{xy}+k\frac{\partial T}{\partial x} \qquad (4d)$$

$$g=u\tau_{xy}+v\tau_{yy}+k\frac{\partial T}{\partial y}; \qquad (4e)$$

p is pressure, ρ is density, e is total internal energy, u and v are velocity components, μ is viscosity, T is temperature, and k is the heat transfer coefficient.

For a general coordinate system the equations are transformed to the following:

$$\frac{\partial \overline{U}}{\partial t}+\frac{\partial \overline{F}}{\partial \xi}+\frac{\partial \overline{G}}{\partial \eta}=\frac{\partial \overline{F}_V}{\partial \xi}+\frac{\partial \overline{G}_V}{\partial \eta} \qquad (5)$$

where,

$$\overline{U}=\frac{\hat{U}}{J},\quad \overline{F}=\frac{\xi_x\hat{F}+\xi_y\hat{G}}{J},\quad \overline{G}=\frac{\eta_x\hat{F}+\eta_y\hat{G}}{J} \qquad (6a)$$

and similarly for the viscous terms,

$$\overline{F}_V=\frac{\xi_x\hat{F}_V+\xi_y\hat{G}_V}{J},\quad \overline{G}_V=\frac{\eta_x\hat{F}_V+\eta_y\hat{G}_V}{J} \qquad (6b)$$

J is the Jacobian of the general coordinate system.

In order to construct the total variation diminishing scheme, the Navier-Stokes equations must be put into the following form,

$$\frac{\partial \overline{U}}{\partial t}+A\frac{\partial \overline{U}}{\partial \xi}+B\frac{\partial \overline{U}}{\partial \eta}=\frac{\partial \overline{F}_V}{\partial \xi}+\frac{\partial \overline{G}_V}{\partial \eta} \qquad (7)$$

where, A and B are the Jacobian matrices for the transformation of $\overline{F}$ and $\overline{G}$ respectively. Therefore A and B can be described as

$$A = \frac{\partial \overline{F}}{\partial \overline{U}} = R_A \Lambda_A R_A^{-1}, \quad B = \frac{\partial \overline{G}}{\partial \overline{U}} = R_B \Lambda_B R_B^{-1} \qquad (8)$$

where the columns of the matrices R_A and R_B are the right eigenvectors of the matrices A and B. Λ_A and Λ_B are vectors which are related to the eigenvalues of A and B, and can be found in reference [9].

3.2 Discretisation of the Equations

The Navier-Stokes equations, in the form shown in equation 5, can be simply discretised into a two step scheme. The first step solves the flow in the ξ-direction, and the second step solves in the η-direction. This method of discretisation can lead to second order accuracy in time.

$$\begin{aligned} \overline{U}^*_{i,j} &= \overline{U}^n_{i,j} - \lambda_\xi\left(\tilde{F}^n_{i+1/2,j} - \tilde{F}^n_{i-1/2,j}\right) \\ \overline{U}^{n+1}_{i,j} &= \overline{U}^*_{i,j} - \lambda_\eta\left(\tilde{G}^*_{i+1/2,j} - \tilde{G}^*_{i-1/2,j}\right) \end{aligned} \qquad (9a)$$

where,

$$\lambda_\xi = \Delta t / \Delta\xi, \text{ and } \lambda_\eta = \Delta t / \Delta\eta \qquad (9b)$$

A global time step, Δt, is calculated with a CFL number of 0.95. The numerical flux functions, F and G, in equation 9a, are calculated using a finite volume approach.

$$\begin{aligned} \tilde{F}_{i+1/2,j} &= \frac{1}{2}\left[\left(\frac{\xi_x}{J}\right)_{i+1/2,j}\left(\hat{F}_{i+1,j} + \hat{F}_{i,j} - 2(\hat{F}_V)_{i+1/2,j}\right) + \left(\frac{\xi_y}{J}\right)_{i+1/2,j}\left(\hat{G}_{i+1,j} + \hat{G}_{i,j} - 2(\hat{G}_V)_{i+1/2,j}\right) + \left(\frac{R_A\Phi_A}{J}\right)_{i+1/2,j}\right] \\ \tilde{G}_{i,j+1/2} &= \frac{1}{2}\left[\left(\frac{\eta_x}{J}\right)_{i,j+1/2}\left(\hat{F}_{i,j+1} + \hat{F}_{i,j} - 2(\hat{F}_V)_{i,j+1/2}\right) + \left(\frac{\eta_y}{J}\right)_{i,j+1/2}\left(\hat{G}_{i,j+1} + \hat{G}_{i,j} - 2(\hat{G}_V)_{i,j+1/2}\right) + \left(\frac{R_B\Phi_B}{J}\right)_{i,j+1/2}\right] \end{aligned} \qquad (10)$$

Note that, the flux functions with a superscript * are calculated using U^*. In reference [1], Yee has offered several other methods of calculating the flux functions, but they are not covered here.

Roe's averaging is applied to cell vertex points in order to calculate the flow variables at (i+1/2,j) and (i,j+1/2).

The vectors, Φ_A and Φ_B, in equation (10) contain the anti-diffusive terms that are under investigation in this report. They provide a second order upwind weighted dissipation term, see reference [1]. The equation for the components of the vector Φ_A are shown here, a similar relation can be formulated for the vector Φ_B.

$$\phi^l_{i+1/2,j} = \sigma\left(a^l_{i+1/2,j}\right)\left(g^l_{i+1,j} + g^l_{i,j}\right) - \psi\left(a^l_{i+1/2,j} + \gamma^l_{i+1/2,j}\right)\alpha^l_{i+1/2,j} \qquad (11)$$

where 'a' denotes the eigenvalues of matrix A, and the components of the vector α can be obtained using,

$$\alpha_{i+1/2,j} = \left(R_A\right)^{-1}_{i+1/2,j}\left(\hat{U}_{i-1,j} - \hat{U}_{i,j}\right) \qquad (12)$$

For the function γ one finds,

$$\gamma^l_{i+1/2,j} = \sigma\left(a^l_{i+1/2,j}\right)\begin{cases} \left(g^l_{i+1,j} - g^l_{i,j}\right)/\alpha^l_{i+1/2,j} & \alpha^l_{i+1/2,j} \neq 0 \\ 0 & \alpha^l_{i+1/2,j} = 0 \end{cases} \qquad (13)$$

The function σ(z) is defined as,

$$\sigma(z) = \tfrac{1}{2}\left[\psi(z) - \lambda z^2\right] \qquad (14)$$

The function ψ(z) is calculated using,

$$\psi(z) = \begin{cases} |z| & z \geq \tilde{\delta} \\ \left(z^2 + \tilde{\delta}^2\right)/2\tilde{\delta} & z < \tilde{\delta} \end{cases} \qquad (15a)$$

This function is introduced to prevent non-physical solutions such as expansion shocks, when one of the eigenvalues goes to zero. The function introduces a small amount of artificial viscosity. A relationship is provided in order that '$\tilde{\delta}$' is suitably scaled for highly skewed grids. This relation is,

$$\tilde{\delta} = \delta\left[\hat{u} + \hat{v} + 0.5c\left(\sqrt{\xi_x + \xi_y} + \sqrt{\eta_x + \eta_y}\right)\right] \qquad (15b)$$

where $\hat{u}$ and $\hat{v}$ are the co-variant velocities. A study of the actual effects of varying the value of the constant, δ, in the solution, appears later in this report. This constant is referred to as the entropy parameter throughout this report.

The term 'g', in equations (11) and (13), is the flux limiter. Five different limiters have been implemented and investigated, they are given in reference [1], and are:

$$g^l_{i,j} = \min\text{mod}\left(\alpha^l_{i-1/2,j}, \alpha^l_{i+1/2,j}\right) \qquad (16a)$$

$$g^l_{i,j} = \left(\alpha^l_{i-1/2,j}\alpha^l_{i+1/2,j} + \left|\alpha^l_{i-1/2,j}\alpha^l_{i+1/2,j}\right|\right)/\left(\alpha^l_{i-1/2,j} + \alpha^l_{i+1/2,j}\right) \qquad (16b)$$

$$g^l_{i,j} = \left\{\alpha^l_{i-1/2,j}\left[\left(\alpha^l_{i+1/2,j}\right)^2 + \varepsilon\right] + \alpha^l_{i+1/2,j}\left[\left(\alpha^l_{i-1/2,j}\right)^2 + \varepsilon\right]\right\}/\left[\left(\alpha^l_{i-1/2,j}\right)^2 + \left(\alpha^l_{i+1/2,j}\right)^2 + 2\varepsilon\right] \qquad (16c)$$

$$g^l_{i,j} = \min\text{mod}\left[2\alpha^l_{i-1/2,j}, 2\alpha^l_{i+1/2,j}, \tfrac{1}{2}\left(\alpha^l_{i-1/2,j} + \alpha^l_{i+1/2,j}\right)\right] \qquad (16d)$$

$$g^l_{i,j} = S.\max\left[0, \min\left(2\left|\alpha^l_{i+1/2,j}\right|, S.\alpha^l_{i-1/2,j}\right), \min\left(\left|\alpha^l_{i+1/2,j}\right|, 2S.\alpha^l_{i-1/2,j}\right)\right]; \qquad (16e)$$

$$S = \text{sign}\left(\alpha^l_{i+1/2,j}\right)$$

The minmod function of a list of arguments is equal to the smallest argument in absolute value if the list of arguments are of the same sign, or is equal to zero if any arguments are of opposite sign.

In equation (16c) ε is included to stop any division by zero. ε is given a small value, usually of the order of 10^{-7}.

The accuracy and shock resolution of Euler solutions increases going from limiters (16a) to (16e). Throughout the rest of this report equations (16a) to (16e) are referred to as Limiters 1 to 5, respectively.

3.3 Grid Generation

The grid used for this problem was calculated using a very simple analytical approach followed by a short smoothing operation. It is necessary to have a high concentration of points at the jet exit, where large changes in the flow will occur, and near the wall, in order to accurately capture the boundary layer.

The grid generator introduces regions of highly concentrated points where specified. This leads to the production of a grid which is highly irregular. Rapid changes in grid point concentration can cause the numerical code to fail, or spurious glitches in the solution of the flow to occur.

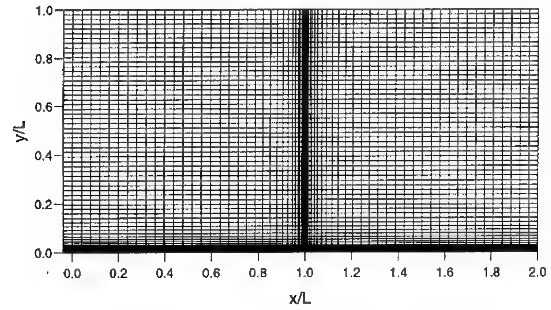


Figure 2. Grid generated for the transverse jet case.

A Laplacian smoothing operator is applied to the grid to solve these problems. Up to 15 smoothing iterations are performed on the grid. The number of iterations executed depends on the degree of irregularity of the grid, and the number of points in the regions where there is a high concentration of points. The grid generated for this test case, using this method is shown in figure 2.

4. RESULTS AND DISCUSSION

The test case for the investigation is a laminar boundary layer developing over a flat plate. A jet issues perpendicularly into the supersonic crossflow from x/L=1, where L=x_{jet}, the position of the jet. The inflow conditions for the flowfield are calculated for a Mach number of 2.61. The inflow conditions of the jet are as follows: the jet pressure ratio, P_j/P_∞=7.0; the temperature ratio, T_j/T_∞=1.0; and M_j=1.0. The Reynolds number of the flow; Re_L=749,000; and the freestream temperature, T_∞=300K. These values were derived from the data provided in reference [3].

4.1 Comparison with Experiment

Other than the results obtained to test the grid dependency of the solution, all of the numerical results produced were calculated on a 100x100 grid as shown in figure 2. The grid has 6 points in the jet, and contains between 30 and 35 points in the boundary layer region. Simple boundary conditions have been used everywhere. The flat plate is modelled using no-slip conditions and is considered to be adiabatic, the inflow condition is fixed at the initial condition, and all outflow conditions use a simple linear extrapolation technique. The jet has a rectangular profile, the actual profile would be closer to a quadratic profile. The jet boundary conditions are fixed at initial conditions for the injected flow.

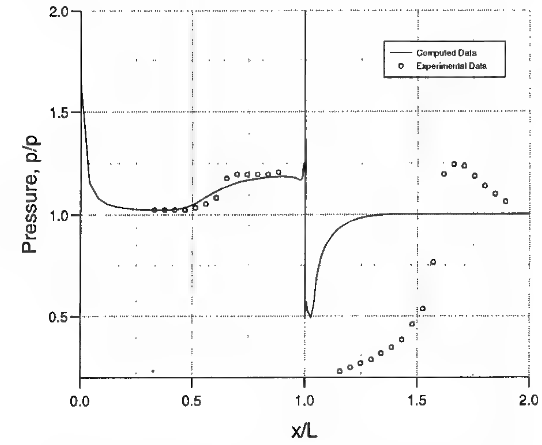


Figure 3. Pressure distribution along the surface of the flat plate.

The pressure distribution along the surface of the plate has been compared with experimental results pruduced by Zukoski et al [3], for the test case described above. The experiments have been done for a three dimensional case, and the jet injection hole was circular. The comparison is shown in figure 3. This numerical solution has been calculated using limiter 1, and with δ=0.001.

Upstream of the jet, the results produced by the code compare well with experiment. Downstream of the jet there is a large discrepancy between experimental and numerical data. This discrepancy is probably due to three dimensional effects which are coming into play around the jet. Some of the flow will be passing around the jet, reducing the mass flow through, and just downstream of it. Another possibility can be attributed to a turbulent region forming just downstream of the jet. At this stage of the work, the numerical code does not include a turbulence model and this part of the flow can not be accurately represented. However, as an initial foray into this problem and for comparing the effects of changing different parameters, the resolution of the results especially near and upstream of the jet is adequate.

4.2 The Entropy Parameter

In order to examine the effects of the entropy parameter, δ, as defined in equation (15), numerical solutions for the test case were found for seven different values of δ, varying from 0.001 to 1. All of the cases, in which the entropy parameter was being studied, were calculated using Limiter 1. This limiter is the most robust one, and was least likely to fail when the more extreme values of the entropy parameter were being tested.

The skin friction and pressure distributions across the flat plate, for different values of δ, are presented in figures 4 and 5, respectively. These plots show the effect of δ on the numerical solution. The pressure plots show that as the entropy parameter is increased the shock wave becomes less well defined. For the highest values of the entropy parameter the shock wave has smeared all the way to the jet injection point. Also by increasing the entropy parameter the low pressure region after the jet, denoting a recirculation region, becomes damped out and the pressure plateau related to the separation region upstream of the jet reduces considerably. The Skin friction distributions show the change in the point of boundary layer separation, x_{sep} (i.e. where the skin friction

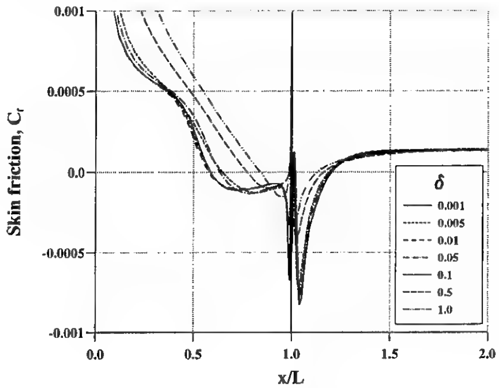


Figure 4. A graph comparing the effects of different values of δ on the skin friction distribution across the flat plate.

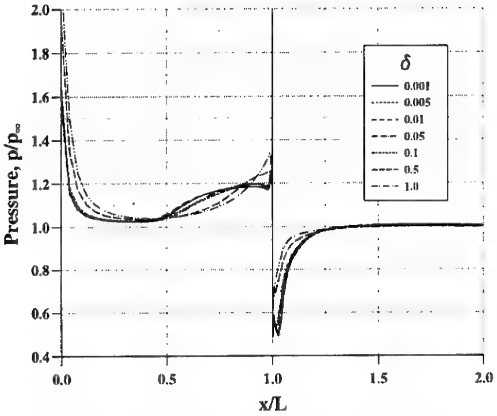


Figure 5. A graph comparing the effects of different values of δ on the pressure distribution along the flat plate.

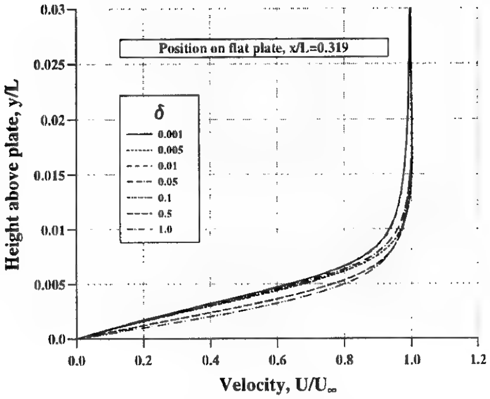


Figure 6. Velocity profiles in the boundary layer for different values of δ.

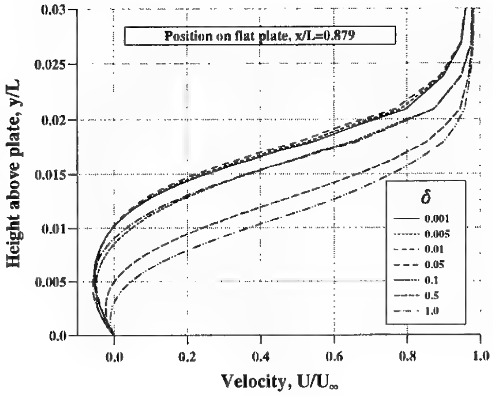


Figure 7. Velocity profiles in the boundary layer for different values of δ.

profile first crosses zero) with δ. As the entropy parameter increases the x_{scp} moves closer to x_{jet}. It can be clearly seen from both of these plots that the choice of δ must be considered carefully.

In essence, changing the value of δ alters the minimum amount of artificial viscosity that is added to the solution. This has the undesirable effect of altering the shape of the boundary layer. Boundary layer profiles plotted for locations, x/L=0.319 and x/L=0.879 are presented in figures 6 and 7. It can be seen from figure 6, in which the boundary layer has not yet separated, that as δ is increased, in general, the boundary layer becomes thicker. The variation is complex, such that the boundary layer thickness reaches a peek, and then it begins to decrease slightly. This is caused by the addition of artificial dissipation to the real viscosity, introduced by the Navier-Stokes equations, in the boundary layer. Figure 7 shows velocity profiles near the wall, after the boundary layer has separated. The change in the boundary layer separation point has a great effect on these profiles. The recirculation region increases and the separated boundary moves further from the flat plate as δ decreases, until δ reaches a value of 0.01. Further lowering of δ moves the separated boundary layer slightly closer to the flat plate. However, it can be seen that, when values of δ below 0.01 are used, the boundary layer profiles seem to become less sensitive to changes in δ, and have converged. .

An examination of the amount of artificial dissipation and real dissipation added to the solution near the flat plate is shown in figures (8a) to (8c). These figures represent dissipation profiles in the boundary layer. The values plotted are the artificial and real second-order terms for the x- and y-momentum parts of $\tilde{G}$ (see equation 10), respectively artificial G_2, real G_2, artificial G_3 and real G_3. These terms have been used because high gradients of variables are expected normal to the wall direction. All of the graphs show the amount of artificial dissipation added to the scheme compared to the real dissipation. Not surprisingly, the addition of the artificial dissipation seems to cause a considerable change in the real dissipation profile which also must have an effect on the boundary layer thickness. Figures 8a-1 to c-1 are plots for the unseparated boundary layer. They show that even for small values of δ, the amount of artificial dissipation added to the y-momentum equation is considerable compared to the real viscosity. For the x-momentum equation the artificial dissipation is comparatively small for low values of δ, but it increases to an overwhelming amount for δ=1.0. Figures 8a-2 to c-2 show the dissipation profiles for a separated boundary layer. These plots show that even for very low values of the entropy parameter, the artificial viscosity is still high enough to be interfering with the solution. Artificial dissipation has been introduced by the scheme to sharpen the definition of the boundary layer as if it was a shock wave. Obviously this isn't necessary in the case of the boundary layer. However, the effect of increasing δ seems to be an increase in the thickness of the separated boundary layer.

Many of the methods used to limit the introduction of the artificial dissipation in the boundary layer would not work in the separated region. Caughey and Varma [6] provide two methods of limiting the artificial dissipation added by the scheme. The first method simply sets the artificial dissipation on surfaces to zero. The other method scales the dissipation using a function of the local Mach number. Neither of these methods are likely to have a great effect on limiting the artificial dissipation term, as the artificial dissipation being added in the separated region is neither near the flat plate, nor in a relatively low velocity region. Also, for most values of the entropy parameter, the artificial dissipation added at the wall is relatively close to zero.

Figures 9a to c give a more qualitative view of the artificial and real dissipation being added near the wall. Contour plots

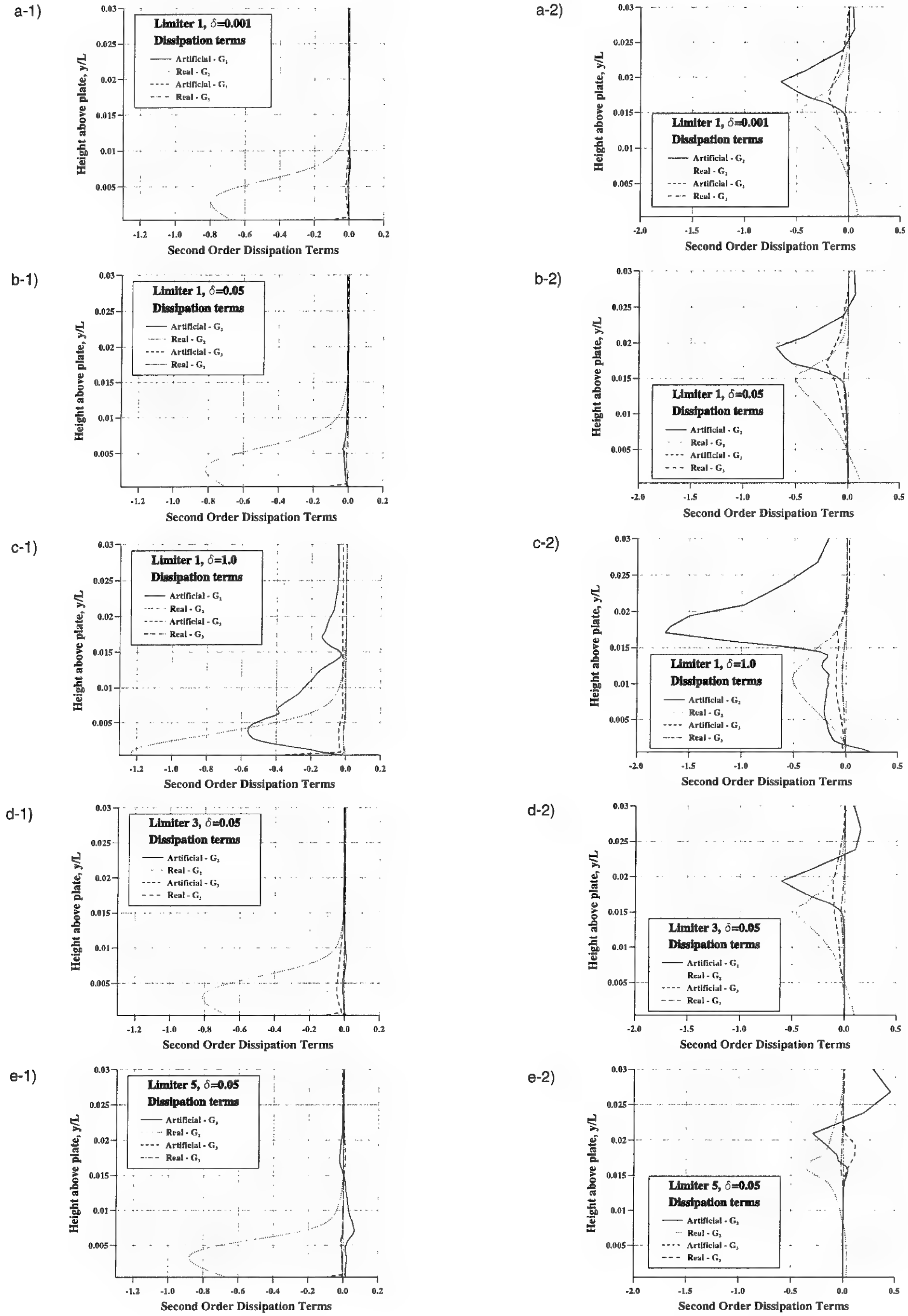


Figure 8. A comparison of dissipation profiles, for different values of δ and different limiters, through an unseparated (a-1 to e-1) and separated (a-2 to e-2) boundary layer.

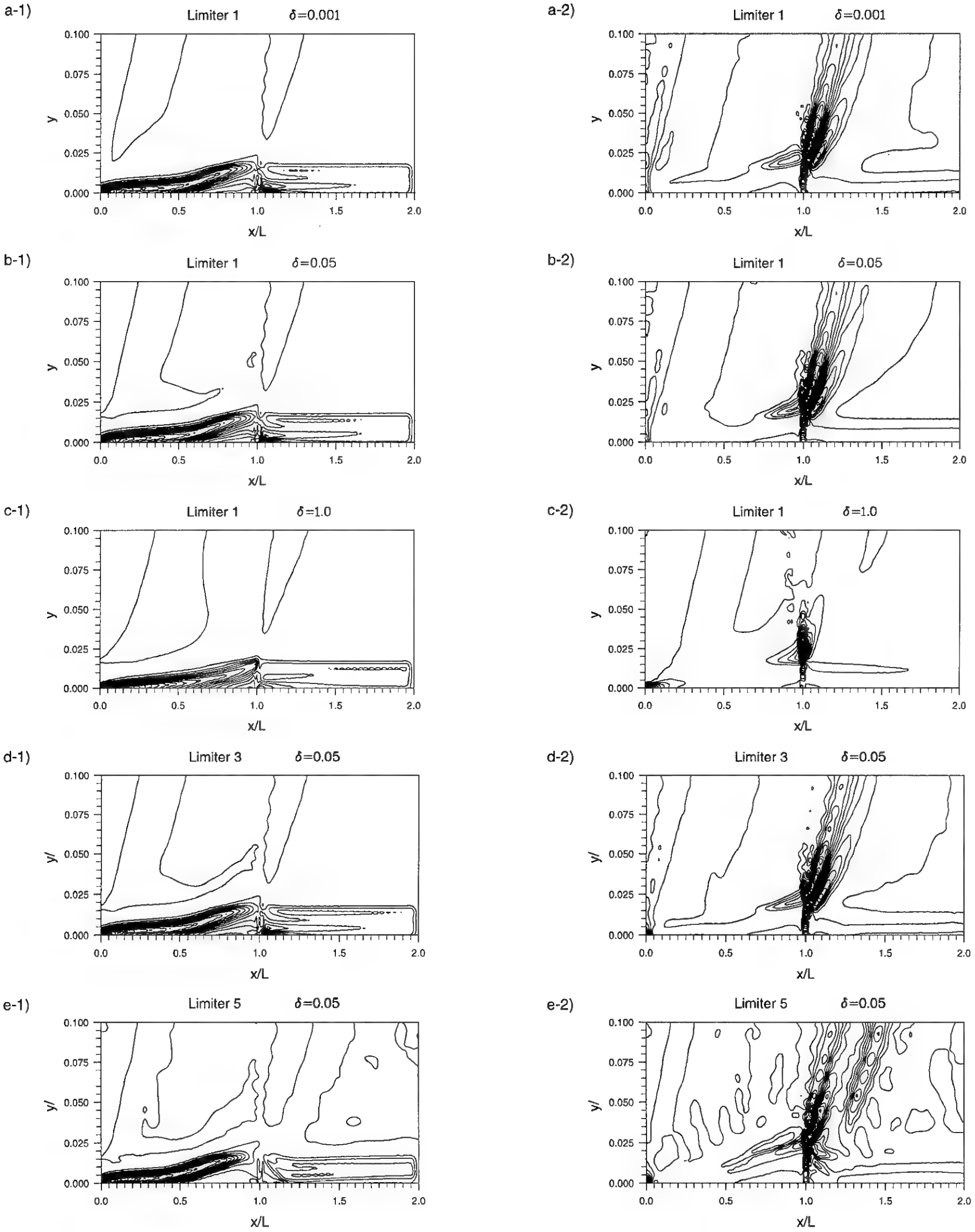


Figure 9. Contour plots of real dissipation (a-1 to e-1) and artificial dissipation (a-2 to e-2) for the flowfield in the vicinity of the flat plate using several different limiters and values of δ.

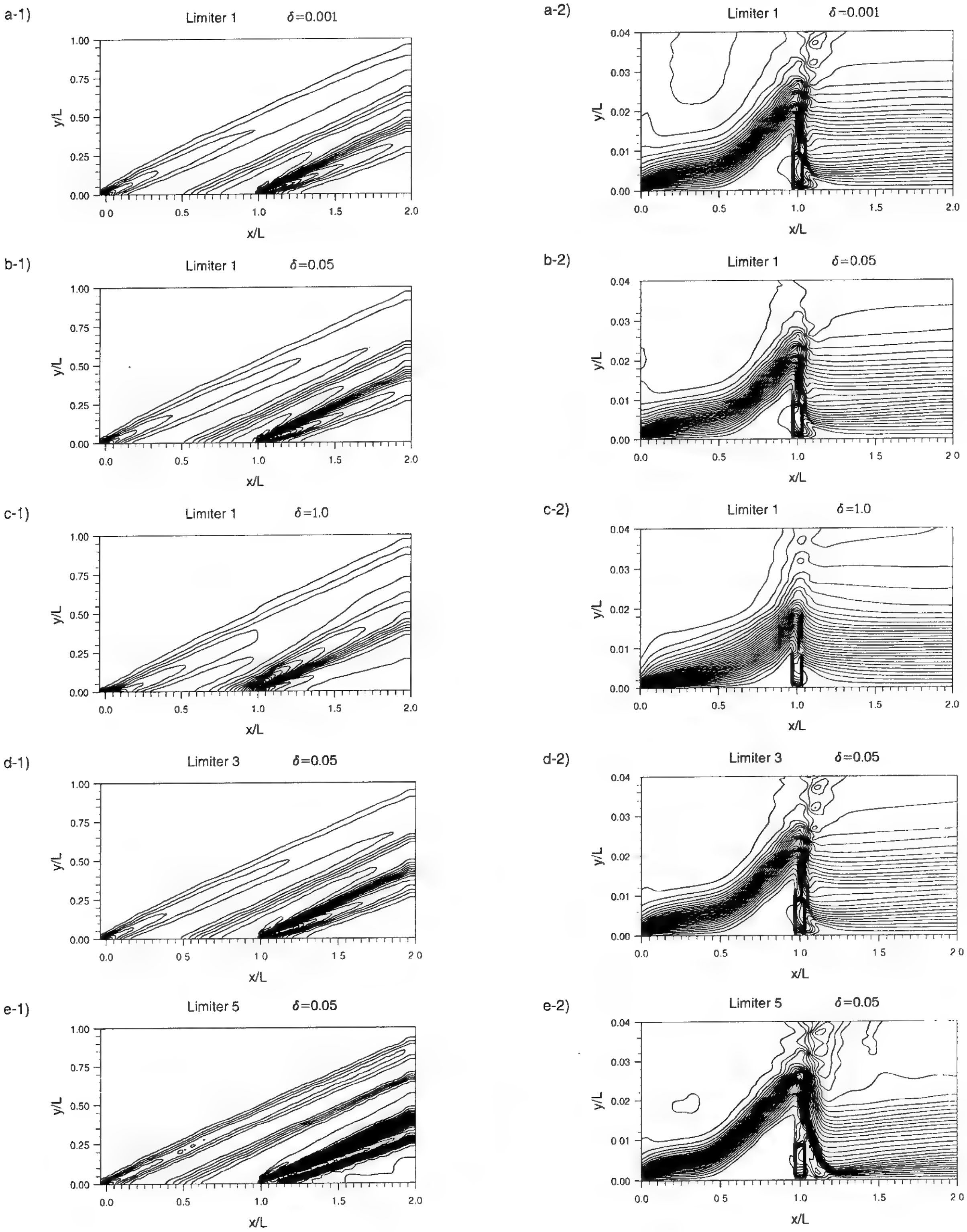


Figure 10. Contour plots of pressure across the whole flowfield(a-1 to e-1) and Mach number near the flat plate(a-2 to e-2) using several different limiters and values of δ.

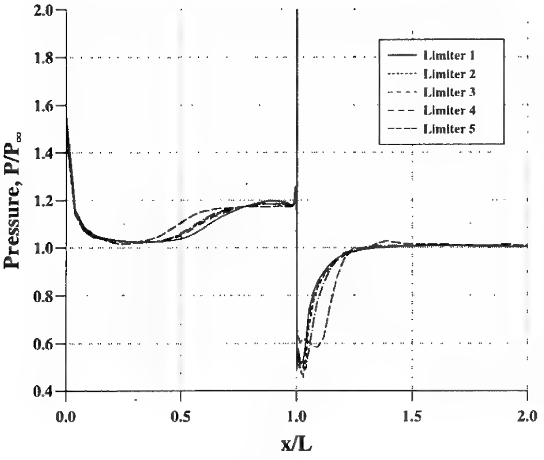


Figure 11. Pressure distribution along the flat plate for five different limiters.

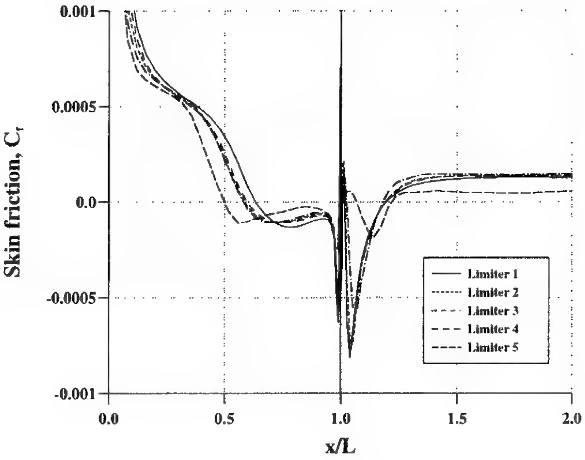


Figure 12. Skin friction plots for the flat plate, for each limiter.

of G_2-real and G_2-artificial for three values of δ have been included. These plots can not be directly compared with each other as the variation of the contour lines is different for each case. They show general trends only. Of particular interest is the small region just upstream of the jet where both the artificial and real dissipations are being added in comparable amounts.

Figures 10a-1 to c-1 show pressure contours for the whole flowfield. These plots illustrate the effects of increasing δ on the whole solution. The separated shock wave becomes less well defined until it is not even clear that there is a shockwave, and the shock wave caused by the reattachment of the boundary layer, downstream of the jet, also becomes less easy to recognise.

Figures 10a-2 to c-2 are contour plots of Mach number for the flow near the wall. These plots illustrate the Mach disk, the separation and reattachment of the boundary layer, and regions of recirculation. The plots also show the slight increase in boundary layer thickness with increasing δ. The small pocket of recirculation just downstream of the jet is reduced in size when large values of δ are used.

In conclusion, when δ is increased to a value of 0.5 and beyond, the solution becomes highly inaccurate. However, for values of the entropy parameter equal to and below 0.01, the solution seems to be less sensitive to changes in δ. This profile represents the solution where very little artificial dissipation is being added to the boundary layer and it is also the solution closest to the experimental pressure data.

Other methods of modelling the entropy parameter, see equation (15b), have been defined by Müller [10,11] and Lin [2]. Müller has used an entropic function of the local spectral radii to model the entropy parameter. A brief examination of this technique showed that for δ=0.005, and using limiter 1, the results produced were very similar to the results produced using the method described by equation (15b). It is possible that Müller's entropy function will be more effective when used with other limiters, and this will be investigated in later work.

Lin [2] states that the viscous flow results using the scheme described in this investigation are unacceptable if a value of δ=0.25 is used. This agrees with the results presented here. Lin suggests using a form of the entropy function which is similar to that shown in equation (15b). Lin also uses different values of the entropy parameter for the linear and non-linear waves. This means a much smaller value for the entropy parameter can be used for the linear waves and hence the boundary layer will be less affected by the artificial dissipation term. An investigation of how effective this concept is, is underway. The results presented by Lin are encouraging.

4.3 Comparing Limiters

The choice of flux limiter is an important factor in TVD schemes. Five different limiters have been investigated here. Limiter 1 is the well known minmod limiter. Limiter 2 is the limiter formulated by Van Leer and limiter 5 is known in the literature as the "Roe's Superbee" and is highly compressive.

It was found that the numerical scheme failed when limiter 5 was used if the entropy parameter was set below a value of 0.05. Therefore in order that the limiters could be compared, the computed solution was found for each limiter with the entropy parameter set at a value of 0.05. In all likelihood this means that the best possible results for limiters 2, 3 and 4 have not been found.

Figures 11 and 12 show pressure and skin friction distributions along the flat plate for each of the limiters. The pressure distributions show that the major effect of using different limiters is to change the point of boundary layer separation. The shock definition on the surface of the flat plate, denoted by the pressure gradient of the shock wave, is not greatly improved by using a more compressive limiter.

From figures 11 and 12 it can be seen that limiters 2, 3 and 4 produce similar results. Also, these results compare well with the results obtained using limiter 1 with values of δ below 0.01, as shown in figures 6 and 7, but generally it seems limiters 2, 3 and 4 are better than limiter 1.

From figure 12, one can see that limiter 5, the "Superbee," does not seem to be well conditioned for this problem, and is probably unacceptable for use when solving any viscous flow problem. The corresponding skin friction distribution is considerably different from those obtained by the other limiters, both upstream and downstream of the jet.

Figures 13 and 14 present velocity profiles for each of the limiters. The plots given in figure 13 are for the unseparated

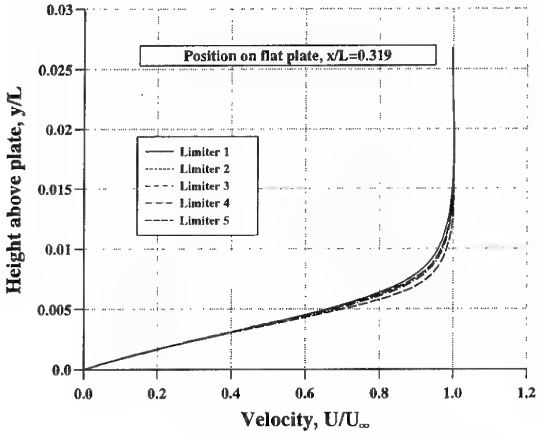


Figure 13. Velocity profiles in the unseparated boundary layer for different limiters.

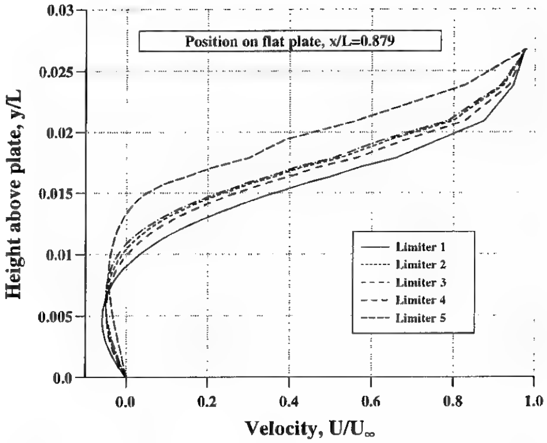


Figure 14. Velocity profiles for the separated boundary layer for different limiters.

boundary layer. As expected, the data suggests that limiters with a more compressive nature reduce the thickness of the boundary layer. In the case of the scheme using limiter 5, the boundary layer thickness has been significantly reduced. Again, limiters 2, 3 and 4 produce similar profiles. Limiter 1 produces a slightly thicker boundary layer than the others.

The results in figure 14 are velocity profiles for the boundary layer after it has separated. Limiter 2, 3 and 4 produce similar profiles, however there is now a more significant degree of difference in the results being obtained. The variation in boundary layer thickness for different limiters, is reversed compared to the unseparated case. The more compressive limiters produce a thicker boundary layer. Primarily this is caused by the change in x_{sep} for the different limiters. These results are similar to those obtained for low values of the entropy parameter using limiter 1, as shown in figure 7.

Dissipation profiles for limiters 1, 3 and 5 shown in figures 8b, d and e respectively, indicate that for the unseparated boundary layer the artificial dissipation introduced into the y-momentum equation is high for limiter 3, but low for the other two limiters. Limiter 5 introduces artificial dissipation of the opposite sign to the real dissipation, in the x-momentum equation. This rèsult explains the decrease in the boundary layer thickness when using limiter 5. Also the amount of real dissipation being added when limiter 5 is being used is slightly higher than for the other limiters.

The dissipation profiles for the separated boundary layer show a marked difference between each of the limiters used. The amount of real dissipation being added when limiter 5 is being used is much smaller than that added when the other limiters are used. The artificial dissipation being added to the x-momentum also seems to be reduced, although there is a region at the edge of the boundary layer where large amounts of dissipation of the opposite sign is being added. Each of the other limiters also add this opposing dissipation but not in such a comparably vast quantity. However the other limiters add more dissipation in other regions of the boundary layer, e.g. near the wall. Also in contrast to limiters 1and 3, the artificial dissipation added to the y-momentum equation by limiter 5 is of the opposite sign to the real dissipation being added. These graphs go some way towards explaining why the results from limiter 5 are so different to the results obtained from the other limiters.

The contour plots of dissipation for limiter 1, 3 and 5 are shown on figures 9b, d and e. Two main conclusions can be drawn from these graphs. Firstly, on the real dissipation plots, the concentration of the contours uptstream of the jet are far higher than for the other limiters, and there is a small region of concentrated contours downstream of the jet for limiters 1 and 3 which is not as prominent for limiter 5. Secondly, the artificial dissipation added by limiter 5 is quite different in pattern to the dissipaton added by the other limiters.

The pressure contours shown in figure 10, show how the more compressive limiters produce better defined shocks waves and expansion fans. The movement of the separation shock wave further upstream of the jet, for limiter 5, is also clearly illustrated here. The Mach number contours show the change in the thickness of the boundary layer for different limiters. The plot for limiter 5, most clearly illustrates the reattachment of the boundary layer. The recirculation region downstream of the jet is much bigger for limiter 5, than for the other two limiters present.

The results presented here show that limiter 5 is not a good choice of limiter for a viscous flow problem, because of its highly compressive nature. Limiter 1 produces acceptable results if the value of the entropy parameter is limited to 0.01 for this test case. Limiters 2, 3 and 4, the mid-range limiters including the Van Leer limiter, produce similar results and are probably best suited for this problem.

4.4 Effects of Grid Size

In order to check the dependence of the solution on the available number of grid points, the numerical code was run on three different grids with different point concentrations. The grid sizes used for this study were, 50x50, 100x100 and 200x200. These calculations were all done using Limiter 3 and the entropy parameter, δ=0.05.

A comparison of the pressure distribution along the flat plate for each grid is shown in figure 15. The plots show a severe degradation of results between the different meshes used. The most coarse grid does not define the shock wave well, and the pressure valley and plateau upstream of the jet do not reach the values produced by the other two grids. The main differences between the 200x200 and the 100x100 grid are the improved shock definition at the wall, and a slight increase in the pressure plateau for the higher grid concentration.

The skin friction distribution comparison given on figure 16, show a marked difference in profile for each grid type. The skin friction given by the 200x200 grid is far higher than the

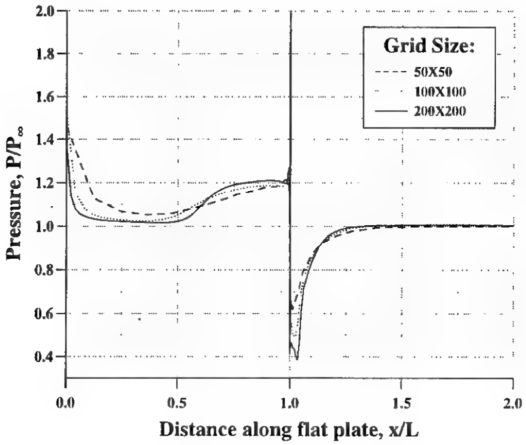


Figure 15 A graph showing the effects of grid size on the pressure distribution along a flat plate.

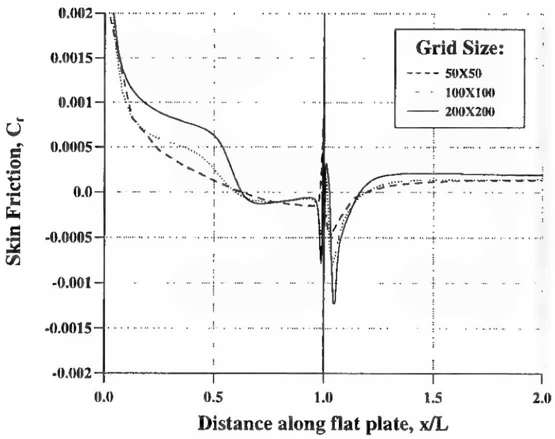


Figure 16 A graph showing the skin friction profile along the flat plate for different grids.

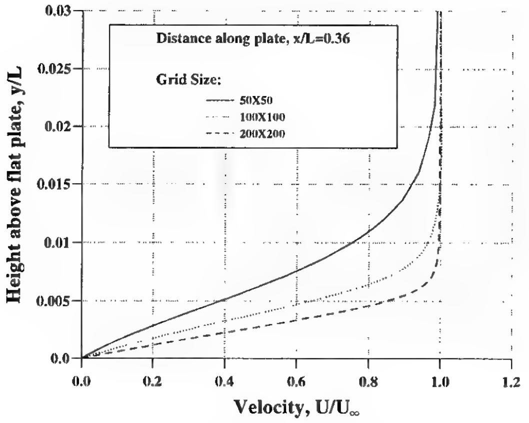


Figure 17. Velocity profiles for an unseparated boundary layer for different grids.

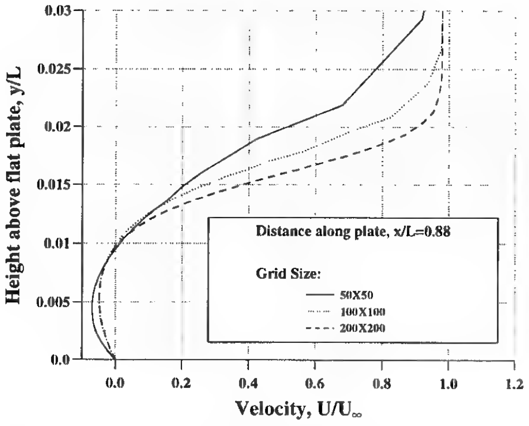


Figure 18. Velocity profiles for a separated boundary layer for different grids.

other grids, and the recirculation region just downstream of the jet is much better defined when more points are used. The differences can be attributed to the fact that, for a higher point concentration the boundary layer is better defined. The location at which the boundary layer separates does not vary greatly over a change in grid point concentration.

Boundary layer velocity profiles are provided on figures 17 and 18 for each of the grids. The unseparated boundary layer profiles, presented in figure 17, show the simple improvement in boundary layer definition as more points are put near the wall. Table 1 gives the number of points in the boundary layer for each case. The grid used earlier in the report does not provide an accurately defined boundary layer, but it was of acceptable quality for the comparative study being performed.

Figure 18 shows velocity profiles in the separated region for each of the grids. In the recirculation region near the wall, the 100x100 and 200x200 grids produce similar results. The 50x50 grid produces a solution with a larger amount of recirculation than the other two grids.

Both sets of velocity profiles show that the grid concentration has a large effect on the calculation of the boundary layer, and hence the rest of the solution.

Figures 19 and 20 show the artificial dissipation terms added to the x-momentum equation profiled through the unseparated and separated boundary layer, respectively. For the unseparated boundary layer, the 100x100 and 200x200 grid produce a similar profile, and near the wall the amount of artificial dissipation being added for each of these cases is quite close. The real differences in the profiles occur towards the edge of the boundary layer. The amount of dissipation provided by the scheme using the 50x50 grid is sometimes twice as much as that introduced when the other grids are being used.

In the case of the separated boundary layer, the 50x50 grid seems to add less artificial dissipation than the other two grids. Although it is not clear from figure 20, the dissipation being added by the 100x100 and 200x200 grid is almost identical in

GRID SIZE	Points in the boundary layer
50X50	23
100X100	32
200X200	43

Table 1. Grid points in the boundary layer.

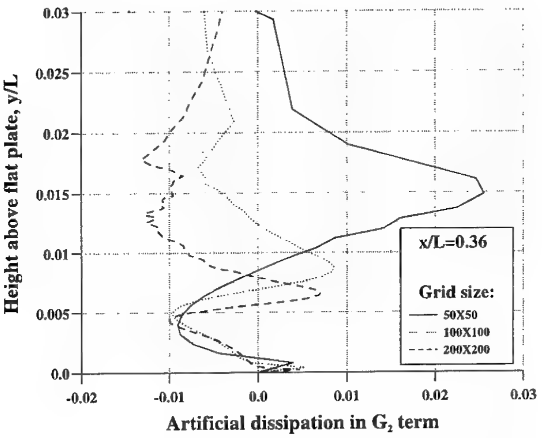


Figure 19 Dissipation profiles through an unseparated boundary layer for different grids.

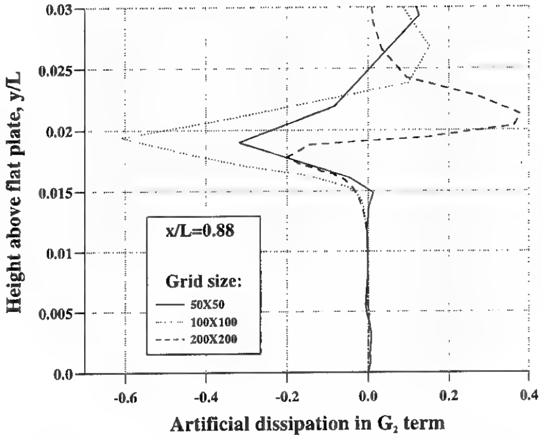


Figure 20 Dissipation profiles through a separated boundary layer for different grids.

the recirculation region. It is only when the edge of the separated boundary layer is reached that large differences become apparent. In the region of the separated boundary layer, the 100x100 grid seems to add far more dissipation in this region than the other two grids. It can be clearly seen that in this region the grid is becoming more coarse, and numerical truncation errors are becoming dominant.

It is clear from this brief study, that grid point density has a great effect on the viscous regions of the solution, and the amount of artificial dissipation being added by the numerical scheme should not be analysed in isolation.

5. CONCLUSIONS

The numerical scheme described in this study has been used to model a transverse jet interacting with a supersonic flow. This test case differs from others that have been used to study artificial dissipation, as the problem includes a separated boundary layer, and reverse flow regions.

This scheme can be used to produce adequate viscous supersonic flows. It has been shown here, that the choice of limiter and value for the entropy parameter is of great importance for producing good results. For the transverse jet test case, limiters 2, 3 and 4 produces good results, and limiter 4 is probably best suited. Although the examination of the entropy parameter was not done for each limiter separately, when limiter 1 was used, it was found that the best results were produced when the entropy parameter was no greater than 0.01. It is reasonable to assume that this range of values will produce acceptable results with the other limiters.

6. REFERENCES

1. Yee, H.C., Klopfer, G.H., Montagne, J.L. "High-Resolution Shock-Capturing Schemes for Inviscid and Viscous Hypersonic Flows", J. Comput. Phys., **88**, 1990, pp31-61.
2. Lin, H-C. "Dissipation Additions to Flux-Difference Splitting", J. Comput. Phys.,**117** ,1995, pp20-27.
3. Zukoski, E.E., Spaid, F.W. "Secondary Injection of Gases into a Supersonic Flow", AIAA J., **2**, Oct 1964, pp1689-1696.
4. Allmaras, S.R. "Contamination of laminar boundary layers by artificial dissipation in Navier-Stokes solutions", Proceedings of the Conference on Numerical Methods in Fluid Dynamics, Reading, UK, 1992.
5. Tatsumi, S., Martinelli, L., Jameson, A. "Design, Implementation, and Validation of Flux Limited Schemes for the Solution of the Compressible Navier-Stokes Equations", AIAA-94-0647.
6. Caughey, D.A., Varma, R.R. "Evaluation of Navier-Stokes Solutions Using the Integral Effect of Numerical Dissipation", AIAA J., 32, Feb 1994, pp294-300.
7. Turkel, E., Vatsa, V.N. "Effect of Artificial Viscosity on Three-Dimensional Flow Solutions", AIAA J., **32**, Jan. 1994, pp39-45.
8. Swanson, R.C., Turkel, E. "Aspects of a High-Resolution Scheme for the Navier-Stokes Equations", AIAA-93-3372-CP.
9. Yee, H.C., Kutler, P. "Application of Second order Accurate Total Variation Diminishing Schemes to the Euler Equations in General Geometries", NASA TM-85845, 1985.
10. Müller, B. "Implicit Upwind Finite Difference Simulation of Laminar Hypersonic Flow over Flared Cones", Notes on Numerical Fluid Mechanics, **29**, 1990.
11. Müller, B. "Comparison of Upwind and Central Finite Difference Methods for the Compressible Navier-Stokes Equations", Notes on Numerical Fluid Mechanics, **30**, 1991.

A Flux Filter Scheme Applied to the Euler and Navier Stokes Equations

A.Vinckier
J. Jacobsen
S. Wagner
Institut für Aerodynamik und Gasdynamik
Universitat Stuttgart
Pfaffenwaldring 21
70550 Stuttgart, Germany

1 SUMMARY

In this contribution, we present a multi-dimensional upwind scheme. In contrast to the Flux-Vector or the Flux-Difference-Splitting method, where an upwind operator is used before the residual is calculated, this scheme uses an operator on the discrete flux integration or flux balance and assigns then filtered parts of the residuals to the vertices of a cell.

The so called Flux-Filter operator, will be derived in a consequent manner on an one-dimensional basis with the purpose to allow a stable updating. The scheme is linearity preserving and should therefore lead to an improved accuracy.

The Flux-Filter scheme has been successfully implemented on the Euler and Thin Layer Navier Stokes equations, for structured and unstructured grids. The unstructured grids are made of triangular and quadrilateral cells.

List of Symbols

A	flux jacobian matrix
$D^{(2)},D^{(4)}$	artificial viscosity vector
E	energy
F	inviscid flux
I	identity matrix
P	pressure source vector
$R^{(p)}$	flux residual with p order of integration
S	surface
U	conservative vector
X	eigenvector matrix
a,b	convection velocity
p	pressure
q	preferential factor
t	time
u	velocity
$u_{i,j}^{n}$	scalar value on point i,j at time level n
$\mathcal{F}$	flux-filter matrix
Ω	volume
α	coefficient
δ	kronecker delta
λ	eigenvalue
θ_x,θ_y	CFL number in x und y direction
σ	root of scheme
ω	root of spacial discretization

2 INTRODUCTION

Upwind Methods have become very popular over the last decade and can be categorized into two major methods, the Flux-Vector-Splitting and the Flux-Difference-Splitting method.

The schemes of Steger&Warming [1] and Van Leer [2] are representative for the Flux-Vector-Splitting schemes. Here, the fluxes are splitted into two parts, a positive and negative part. The positive, respectively negative, fluxes have purely positive, respectively negative, eigenvalues and can therefore be differenciated with backward, respectively forward upwinding.

The Flux-Difference-Splitting methods or Riemann solvers are another group of schemes. Here the conservative variables are taken to be piecewise constant between the cell faces. At the faces there is a fluid state on the left side and a different fluid state at the right side, which results in an interaction. This interaction, seen in one dimension, has a mathematical and physical exact solution. It is equivalent to the shock tube problem also known as the Riemann problem. The most popular approximated Riemann solvers are from Roe [3] and from Osher [4].

Those schemes solve the upwinding by treating each space dimension seperately along the gridlines or along the normals of the cell faces. This has a disadvantage that contact discontinuities which are not aligned with the grid are not properly solved. To overcome this problem, a new group of methods has emerged since the early 1990's, the so-called multi-dimensional upwinding scheme. Two distinct methods have been developed up to now; the Flux-Function methods [5] and the Flux-Fluctuation methods. The Flux-Function scheme will calculate the flux through a face in a Riemann manner but independently from the

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

grid. The variation in Flux-Function methods lays in the 'Riemann' directions chosen which mainly consist of the convection and pressure wave directions in contrast to the main grid directions. The Flux-Fluctuation methods are based on the flux integration upon a upwinding method distributes the residuals values to the cell's vertices.

The Flux-Filter scheme is a variant of the Flux-Fluctuation methods. Similar approaches are developed by Rossow [6], Giles [7] etc., but this scheme differs in the way how the flux residual is calculated and distributed to the vertices. The distribution is based on the characteristical propagation directions along the grid lines. The Flux-Filter is an operator which selects those quantities of a flux residual that propagate towards the cell vertex of interest. This scheme can be applied to structured and unstructered grids and can solve the Euler and Navier Stokes equations. Stability analysis has shown that preferential direction flux integration and artificial viscosity is required. In case of the structured grids a combined second and fourth order viscosity is implemented, for the unstructured grid method only a second order artificial viscosity have been introduced.

In the following section the basic idea of the Flux-Filter scheme is explained for the quasi one-dimensional Euler equations, followed by the extension to two dimensions. An analysis of the scalar model equations will highlight some problems associated with the trapezoidal flux integration. In the last section a number of results will be presented.

3 ONE DIMENSIONAL FLUX FILTER SCHEME

The discrete quasi 1-D Euler equations for a cell which is located between the grid points i and $i-1$ is given by

$$\frac{\Delta SU}{\Delta t}+\left[\frac{F_i-F_{i-1}}{\Delta x_{i-1/2}}-P_{i-1/2}\right]=0 \qquad (1)$$

where U is the conservative solution vector, F the flux vectors and S the cross sectional area.

$$U=\begin{bmatrix}\rho\\ \rho u\\ E\end{bmatrix}\quad F=\begin{bmatrix}\rho u\\ \rho u^2+p\\ (E+p)u\end{bmatrix}\quad P=\frac{dS}{dx}\begin{bmatrix}0\\ p\\ 0\end{bmatrix} \qquad (2)$$

The residual $\frac{\Delta SU}{\Delta t}$ needs to be distributed over the grid points i and $i-1$. The question is what part of the residual must be imposed on the left and on the right grid-points. The distribution is obtained through an operator $\mathcal{F}$ which must be linear preserving. Hence, the equation for a grid point i is given by

$$\begin{aligned}&\frac{\Delta SU_i}{\Delta t}+\mathcal{F}^+\left[\frac{F_i-F_{i-1}}{x_i-x_{i-1}}-P_{i-1/2}\right]+\\ &\mathcal{F}^-\left[\frac{F_{i+1}-F_i}{x_{i+1}-x_i}-P_{i+1/2}\right]=0\end{aligned} \qquad (3)$$

3.1 The Flux-Filter Operator

The primary purpose of the Flux-Filter operator is to extract those elements of the residual which allow a stable updating [8]. The solution of a numerical flow problem is reached when the numerical process has converged, which can be seen as a numerical equilibrium state. On a local scale in CFD the changes imposed by the flux balance or residual must result in the reduction of the discrepancy in the flux balance. The flux balance for a cell $i-1/2$ (between grid points $i-1$ and i) is given by

$$\frac{\Delta t}{\Delta x}\left(F_i-F_{i-1}\right)=-\Delta U \qquad (4)$$

and suppose the flux balance for cell $i+1/2$ is satisfied, hence

$$\frac{\Delta t}{\Delta x}\left(F_{i+1}-F_i\right)=0\,. \qquad (5)$$

The equation which imposes the condition that the summing up of a filtered portion of the residual on grid point i reduces the discrepancy in flux balance is

$$\begin{aligned}&\frac{\Delta t}{\Delta x}\left(F(U_i+\mathcal{F}(\Delta U))-F(U_{i-1})\right)=\\ &-\left(\Delta U-\alpha\Delta U\right)\end{aligned} \qquad (6)$$

The influence of $\mathcal{F}(\Delta U)$ on equation (5)is

$$\frac{\Delta t}{\Delta x}\left(F(U_{i+1})-F(U_i+\mathcal{F}(\Delta U))\right)=-\alpha\Delta U \qquad (7)$$

The stability demands that for all grid points the discrepancies disappear, or at least remain bounded by ΔU . Hence the stability conditions are

$$0\le\alpha\le 2 \qquad (8)$$

for equation (6) and

$$-1\le\alpha\le 1 \qquad (9)$$

for equation (7). Both conditions lead to the requirement

$$0\le\alpha\le 1 \qquad (10)$$

The Euler equations have the property that $F(U)=A(U)U$ where $A=\partial F/\partial U$. Rewriting equation (6) and assuming ΔU to be small leads to

$$\frac{\Delta t}{\Delta x}\left(A(U_i)(U_i + \mathcal{F}(\Delta U)) - F(U_{i-1})\right) \\ = -\Delta U + \alpha \Delta U \tag{11}$$

Subtracting equation (11) with equation (4) gives

$$\frac{\Delta t}{\Delta x} A_i\ \mathcal{F}(\Delta U) - \alpha \Delta U\ = 0 \tag{12}$$

The Jacobian matrix A possesses a complete set of real eigenvectors, hence

$$A = X \Lambda X^{-1} \tag{13}$$

where X is the eigenvector matrix and Λ is the eigenvalue matrix. The operator $\mathcal{F}$ is now defined as

$$\mathcal{F} = X I^{\delta} X^{-1} \tag{14}$$

where I^{δ} is a diagonal matrix with 0 or 1. Hence

$$\left(\frac{\Delta t}{\Delta x} X \Lambda X^{-1}\ X I^{\delta} X^{-1} - \alpha I\right) \Delta U = 0 \tag{15}$$

The solution of this eigenvalue problem is

$$\alpha_i = \frac{\Delta t}{\Delta x} \lambda_i \delta_i \tag{16}$$

Imposing condition (10) on all eigenvalues gives

$$0 \leq \frac{\Delta t}{\Delta x} \lambda_i \delta_i \leq 1 \tag{17}$$

Therefore, to obtain a stable scheme the δ_i must be zero when λ_i is negative and $\frac{\Delta t}{\Delta x}(\lambda_i \delta_i)_{\max} \leq 1$. The first condition defines the Flux Filter Operator:

$$\mathcal{F}^{\pm} = X I^{\pm} X^{-1} \tag{18}$$

where I^{+} is a diagonal matrix where the elements are 1 where the corresponding eigenvalue λ_i are positive, analogue for I^{-}. The second constraint imposes the well-known CFL condition for an explicit scheme.

3.2 Implementation of the Flux-Filter operator

The previous section has defined an operator which will theoretical allow a stable iterations process. For its implementation three facts must be taken into account:

1. The additional term P.
2. The distribution of the residual may not lead to any sources or sinks.
3. The filters may not block numerically the propagation of information.

The introduction of the P-vector should require a reformulation of the filter based upon the eigenvalues and eigenvectors of $\frac{F}{\Delta x} + P$. The determination of those eigenvalues and eigenvectors will lead to a severe increase in the numerical workload and therefore the Flux-Filter will be based on the eigensystem of the flux.

In order to assure the conservation of the scheme, the introduction of the filter operators may not lead to additional sources, hence

$$\mathcal{F}^{+}(U_a)\left[\frac{F_i - F_{i-1}}{\Delta x_{i-1/2}} - P_{i-1/2}\right] \\ + \mathcal{F}^{-}(U_b)\left[\frac{F_i - F_{i-1}}{\Delta x_{i-1/2}} - P_{i-1/2}\right]. \\ = \left[\frac{F_i - F_{i-1}}{\Delta x_{i-1/2}} - P_{i-1/2}\right] \tag{19}$$

where the positive, respectively negative, Flux-Filter is set by a still undefined U_a, respectively U_b. This implies that

$$\mathcal{F}^{+}(U_a) + \mathcal{F}^{-}(U_b) = I \tag{20}$$

which is always true when U_a is identical to U_b. The obvious choices are

$$U_a = U_b = \frac{(U_i + U_{i-1})}{2} \tag{21}$$

This leads to the Flux-Filter scheme for the quasi one dimesional Flux-Filter scheme:

$$\frac{\Delta S U_i}{\Delta t} + \mathcal{F}^{+}_{i-1/2}\left[\frac{F_i - F_{i-1}}{x_i - x_{i-1}} - P_{i-1/2}\right] \\ + \mathcal{F}_{i+1/2}\left[\frac{F_{i+1} - F_i}{x_{i+1} - x_i} - P_{i+1/2}\right] = 0 \tag{22}$$

where

$$F^{+}_{i-1/2} = X(U_{i-1/2}) I (U_{i-1/2})^{+} X(U_{i-1/2})^{-1}$$

with $U_{i-1/2} = \frac{U_i + U_{i-1}}{2}$

The trivial solution for steady state solution is

$$\left[\frac{F_i - F_{i-1}}{\Delta x_{i-1/2}} - P_{i-1/2}\right] = 0 \tag{21}$$

for all cells of the computational domain. Unlike the traditional methods, the Flux-Filter formulation leads to the solution of the original discrete quasi one dimensional Euler equations (Fig. 1).

4 TWO DIMENSIONAL FLUX-FILTER SCHEME

In this section, the construction of the two-dimensional Flux-Filter [9] scheme will be outlined. The numerical equation for point *i,j* of structured grid becomes:

$$\frac{\Delta U_{i,j}}{\Delta t}+\mathcal{F}^{--}R_{i+1/2,j+1/2}+\mathcal{F}^{+-}R_{i-1/2,j+1/2} \\ +\mathcal{F}^{++}R_{i-1/2,j-1/2}+\mathcal{F}^{-+}R_{i+1/2,j-1/2}=0 \tag{22}$$

Where $\mathcal{F}^{--}$ is the two-dimensional Flux-Filter, which models the upwinding. The residual $R_{i+1/2,j+1/2}$ is the surface flux integration of the computational cell defined by the points $(i,j),(i+1,j),(i+1,j+1),(i,j+1)$. Each cell, which belongs to point *i,j*, contributes a filtered portion of its residual to the temporal change of point *i,j*. The purpose of the Flux-Filter is to extract this portion of the residuals which allows a stable updating.

4.1 Two-Dimensional Flux-Filter Operator

The two-dimensional Flux Filter is based on the one dimensional operators defined in the last section. The following constructions have been tested:

Analysis has lead to the following requirements for the two-dimensional Flux-Filter:

- for conservation, the sum of all Flux-Filters for one cell must add up to the identity matrix: $\sum \mathcal{F} = I$. Although that for a steady state solution all residuals should disappear, this seems to be an unnecessary requirement. However, at a shock the temporal change becomes zero with residuals which are non-zero. If this constraint is not applied the shock position and strength are incorrect.
- if the cell has a supersonic velocity component pointing away from a grid point then the corresponding Flux-Filter must be the null-matrix. This reflects the perception that in the above case no information can propagate towards this point.

This lead to the following scheme for quadrilateral cells (Fig 6.3):

if $\mathcal{F}^{-}\neq[0]$ and $\mathcal{G}^{-}\neq[0]$ then

$$\bar{\mathcal{F}}^{--}=\frac{1}{4}\left(\mathcal{F}^{-}\left(U,\vec{n}_{\xi}\right)+\mathcal{G}^{-}\left(U,\vec{n}_{\eta}\right)\right) \tag{23}$$

else

$$\bar{\mathcal{F}}^{--}=[0] \tag{24}$$

where U is the averaged flow vector for the cell. This formulation reflects the region of dependency. Similar for $\bar{\mathcal{F}}^{-+}, \bar{\mathcal{F}}^{++}$, and $\bar{\mathcal{F}}^{+-}$

The conservation requirement is imposed with

$$Q=I-\left(\bar{\mathcal{F}}^{--}+\bar{\mathcal{F}}^{-+}+\bar{\mathcal{F}}^{++}+\bar{\mathcal{F}}^{+-}\right) \tag{25}$$

if $\bar{\mathcal{F}}^{--}\neq[0]$ then

$$\mathcal{F}^{--}=\bar{\mathcal{F}}^{--}+Q/n \tag{26}$$

where n is the number of non-null matrices.

The formulation for a triangular cell is

if $\mathcal{F}_{ij}^{-}\neq[0]$ and $\mathcal{F}_{ik}^{-}\neq[0]$ then

$$\bar{\mathcal{F}}_{i}=\frac{1}{3}\left(\mathcal{F}_{ij}^{-}\left(U,\vec{n}_{ij}\right)+\mathcal{F}^{-}\left(U,\vec{n}_{ik}\right)\right) \tag{27}$$

else

$$\bar{\mathcal{F}}_{i}=[0] \tag{28}$$

where U is the averaged flow vector for the cell. Similar for $\bar{\mathcal{F}}_{j}$ and $\bar{\mathcal{F}}_{k}$.

The conservation requirement is imposed with

$$Q=I-\left(\bar{\mathcal{F}}_{i}+\bar{\mathcal{F}}_{j}+\bar{\mathcal{F}}_{k}\right) \tag{29}$$

if $\bar{\mathcal{F}}_{i}\neq[0]$ then

$$\mathcal{F}_{i}=\bar{\mathcal{F}}_{i}+Q/n \tag{30}$$

where n is the number of non-null matrices.

4.2 The Flux Residual

The flux residual or flux balance is the flux integral over the cell's circumference.

$$\frac{\partial U}{\partial t}+\frac{1}{\Omega}\oint \vec{F}\cdot\vec{n}\,dS=0 \tag{31}$$

where Ω is the cell's surface. The first order discrete flux integration based on point i,j for a quadrilateral cell is

$$R_{i+1/2,j+1/2}^{(1)}= \\ \frac{1}{\Omega}\Big[\vec{F}\left(U_{i,j}\right)\cdot\left(\vec{n}S\right)_{i+1/2,j}+\vec{F}\left(U_{i+1,j}\right)\cdot\left(\vec{n}S\right)_{i+1,j+1/2} \\ +\vec{F}\left(U_{i,j+1}\right)\cdot\left(\vec{n}S\right)_{i+1/2,j+1}+\vec{F}\left(U_{i,j}\right)\cdot\left(\vec{n}S\right)_{i,j+1/2}\Big] \tag{32}$$

The second order flux integration or trapezoidal integration is given by

$$R^{(2)}_{i+1/2,j+1/2} = \frac{1}{\Omega}\Big[\vec{F}(U_{i,j})\cdot(\vec{n}S)_{i,j} + \vec{F}(U_{i+1,j})\cdot(\vec{n}S)_{i+1,j} + \vec{F}(U_{i+1,j+1})\cdot(\vec{n}S)_{i+1,j+1} + \vec{F}(U_{i,j+1})\cdot(\vec{n}S)_{i,j+1}\Big] \tag{33}$$

Note the inclusion of the information attached to the vertex $i+1, j+1$. The first order integration violates the criteria which requires that the flux calculation of a cell's face is independent of the cell in consideration. The first order integration has the advantage that the resulting scheme is stable. The second order integration fulfills the latter criteria but it has a stability problem, which will be analyzed and clarified in detail. For those reasons, a blended first and second order integration will be used, and is called the preferential direction integration Given by

$$R^{(q)}_{i+1/2,j+1/2} = pR^{(1)}_{i+1/2,j+1/2} + (1-p)R^{(2)}_{i+1/2,j+1/2} \tag{34}$$

where the calculations have shown that for accuracy and stability the optimum value of q is 0.25.

For reason of stability, artificial viscosity must be introduced, which is of course an unwelcomed feature. The artificial viscosity is a second and fourth order damping given by

$$D^{(2)}_{i,j} = d_2\big(U_{i-1,j} - 2U_{i,j} + U_{i+1,j} + U_{i,j-1} - 2U_{i,j} + U_{i,j+1}\big) \tag{36}$$

and

$$D^{(4)}_{i,j} = d_4 \big(-U_{i-2,j} + 4U_{i-1,j} - 6U_{i,j} + 4U_{i+1,j} - U_{i+2,j} - U_{i,j-2} + 4U_{i,j-1} - 6U_{i,j} + 4U_{i,j+1} - U_{i,j+2}\big) \tag{37}$$

4.3 Thin Layer Navier Stokes

The terms for the viscous fluxes are introduced on the right hand side of equation (22). Hence

$$\frac{\Delta U_{i,j}}{\Delta t} + \mathcal{F}^{--}R^{(q)}_{i+1/2,j+1/2} + \mathcal{F}^{+-}R^{(q)}_{i-1/2,j+1/2} + \mathcal{F}^{++}R^{(q)}_{i-1/2,j-1/2} + \mathcal{F}^{-+}R^{(q)}_{i+1/2,j-1/2} = -\frac{\Delta F_v}{\Delta \xi} + D^{(2)}_{i,j} + D^{(4)}_{i,j} \tag{38}$$

To obtain a meaningfull viscous solution, the influence of the viscous term must exceed the influence of the artificial viscosity in the viscous dominant flow regions.

5 SCALAR MODEL EQUATION

In this section, we will analyze the accuracy and stability aspects of the Flux-Filter scheme based on the model equation. The model equation for the multi-dimensional scheme will be the two dimensional convection equation:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = 0 \tag{39}$$

In this equation a quantity u is convected with a velocity a in the x-direction and a velocity b in the y-direction. The theoretical solution preserves the initial function along the convection direction. The accuracy by which a numerical solution approaches this theoretical solution is fully dependent on the numerical scheme.

Equation (39) represents the numerical model equation for a one-by-one dimensional first order upwind scheme and for the Flux-Filter scheme with the first order flux integration

$$\frac{\Delta x \Delta y}{\Delta t}\left(u^{n+1}_{i,j} - u^{n}_{i,j}\right) = -(a\Delta y + b\Delta x)u_{i,j} + (a\Delta y)u_{i-1,j} + (b\Delta x)u_{i,j-1} \tag{40}$$

for positive values of a and b. The multi-dimensional upwinding scheme makes use of the trapezoidal integration of fluxes. Hence, the numerical model equation is given by

$$\frac{\Delta x \Delta y}{\Delta t}\left(u^{n+1}_{i,j} - u^{n}_{i,j}\right) = -\left(a\frac{\Delta y}{2} + b\frac{\Delta x}{2}\right)u_{i,j} - \left(-a\frac{\Delta y}{2} + b\frac{\Delta x}{2}\right)u_{i-1,j} - \left(a\frac{\Delta y}{2} - b\frac{\Delta x}{2}\right)u_{i,j-1} - \left(-a\frac{\Delta y}{2} - b\frac{\Delta x}{2}\right)u_{i-1,j-1} \tag{41}$$

The numerical behaviour of both methods are analyzed for convection directions of 0, 22.5 and 45 degrees for a delta function on the y-axis. Figure 2 presents the result of a 22.5 degree convection using the first-order integration and using the trapezoidal integration method. The diffusion is dominant for the convection direction of 45 degrees, while for a convection direction aligned with the x-axis the scheme produces the theoretical result. For the reason that the numerical scheme will produce considerable cross-diffusion when the convection direction is not aligned with the grid the first order integration scheme is highly grid dependent.

The second order approach leads to the solutions which matches the theoretical solution for convection directions of 0 and 45 degrees. The solution for the 22.5 degree convection is dispersive. The accuracy levels of both methods are given in the next table where the error is the mean square deviation from the theoretical solution:

$$Error = \sqrt{\frac{1}{n}\sum_{j}\left(u_{I,J} - u_{\text{theory}}\right)^2} \tag{42}$$

Error	first order	second
0	0.00	0.00
22.5	0.16	0.12
45	0.23	0.00

Although the second order flux integration produces better results and is therefore less grid dependent, the transient phase can be extremely chaotic. The following case will clarify a peculiar problem. Setting

$$
\begin{aligned}
&u_{0,j} = 0, \forall j \\
&u_{1,0} = 1 \\
&\alpha = \frac{\theta_y}{\theta_x}
\end{aligned}
\tag{43}
$$

with $\theta_x = a\Delta t / \Delta x$ and $\theta_y = b\Delta t / \Delta y$

the values for all $u_{1,j}$ can be determinated in function of α with a the second order integration scheme

$$
u_{1,j} = -\frac{1-\alpha}{1+\alpha} u_{1,j-1} \tag{44}
$$

hence

$$
u_{1,j} = \left(-\frac{1-\alpha}{1+\alpha} \right)^{J} \tag{45}
$$

For the case that α appraoches zero the scheme will produce an undamped oscillatory profile in the complete domain. This reduces the robustness of the Flux-Filter scheme where even a slight disturbance is immediately transmitted, in an unfavorable manner, throughout the numerical domain. In contrast, the profile for the first order integration scheme is

$$
u_{1,j} = \frac{\alpha}{1+\alpha} u_{1,j-1} \text{ or } u_{1,j} = \left(\frac{\alpha}{1+\alpha} \right)^{j} \tag{46}
$$

which does not have a oscillatory behaviour. The problem can be remedied with the addition of artificial viscosity and/or the use of a preferential integration direction. The preferential integration is a blended form between the trapezoidal integration and the first order integration (Eq.34). Hence the preferential integration is a first order integration:

$$
\begin{aligned}
u_{i,j}^{n+1} - u_{i,j}^{n} = &-(.5+q)(\theta_x + \theta_y) u_{i,j} \\
&-(-(.5+q)\theta_x + (.5-q)\theta_y) u_{i-1,j} \\
&-((.5-q)\theta_x - (.5+q)\theta_y) u_{i,j-1} \\
&-(.5-q)(-\theta_x - \theta_y) u_{i-1,j-1}
\end{aligned}
\tag{47}
$$

For the critical case that $\theta_y = 0$ the profile becomes

$$
u_j = \left(-\frac{.5-q}{.5+q} \right)^{j} \tag{48}
$$

which means that for a positive value of q, u_j reduces exponentially. With this scheme the oscillatory behaviour is contained. The errors from the theoretical solution are given in the next table for the method with artificial viscosity and for the method with the preferential integration.

Error	artificial viscosity	preferential integr. q=0.25	preferential + a.v.
0	0.069	0.000	0.038
22.5	0.097	0.093	0.073
45	0.081	0.115	0.095

Hence, modifications, such as artificial viscosity and preferential flux integration, must be included to stabilize the scheme. The pure second order flux integration scheme will not work.

5.1 Von Neumann Stability Analysis

The results of a Von Neumann stability analysis [10] is presented in figure 5 where the maximum amplification factor is plotted in function of θ_x, θ_y for $q = .25$ and a small amount of artificial viscosity. Stability is obtained when $\theta_x, \theta_y \leq 0.3$ which is the upper limit for the CFL number.

Each time discretization method has its own stability contour wherein the roots of the space discretization must lay. Figures 4 presents the spacial-roots for the Flux-Filter scheme for CFL numbers of 0.3, 0.5, 2.0, 10.0. The conclusion is that the gain for using Runge Kutta is minor and that in theory large CFL numbers can be used with an implicit scheme. However, in practise, the implicit scheme worked for a maximum CFL number of 1.0. The gain of factor 3 is not sufficient to justify the use of an implicit scheme, due to the significant increase in workload.

6 RESULTS

The supersonic wedge

The geometry is a two dimensional channel with a 15° wedge and followed by a 15° expansion corner (Fig 6). The inflow Mach number is set to 2. This configuration induces interactions between shock and expansion waves [11]. A shock wave is produced at the wedge and reflected at the upper boundary. The reflected shock wave is weakened by the expansion fan. The expansion fan is also reflected by the upper boundary. Dependent upon the length of the channel, the shock wave and expansion fan are reflected manifold. Analytical results predict a Mach number of 1.454 behind the first shock wave. The maximum deflection for this Mach number is 10.5°, which implies that the first reflection will induce a subsonic flow with an entropy layer (or slip stream). This problem is a widely used standard test case [11]. The grids are a structured 180 by 60 grid, a unstructured grid with 1664 quadrilateral elements and an unstructured grid with 3633 triagular elements. Figure 6 presents the mach number distribution for the 3 grids.

NACA0012 Transonic

This standard AGARD test case [12] consists of a NACA0012 profile in a transonic flow. The angle of attack is 1.25° and the free stream Mach number is 0.8. The main features are a strong shock located at $x = 0.62$ on the upper surface and a weak shock at $x = 0.37$ on the lower side. The Mach number distributions is given in figures 7.

This problem is solved with a Runge-Kutta scheme on a 160 by 50 structured grid. The maximum CFL number was 0.4. After 3000 iteration steps, a pseudo convergence was obtained, where the upper shock position continued to move back and forward around one grid cell.

The predicted Cl (0.3708) and Cd (0.0213) coefficients correspond well to those given by the AGARD test.[12]

NACA0012 Mach 0.5 Reynolds 5000

This test case [13] demonstrates the use of the Flux-Filter scheme on the Navier Stokes equation. The test case consists of a subcritical flow (Ma=0.5) over a NACA-0012 with a Reynolds number of 5000. The angle of attack is 0 degrees. The flow has a recirculation at the trailing edge. The predicted location of separation from references is at x=0.82 This scheme predicts the flow separation to at x=0.92 (Fig. 8), which indicates that the Flux-Filter scheme has an incorrect degree of dissipation.

7 CONCLUSIONS

The Flux Filter scheme has been applied to the Euler and Navier Stokes equations, on structured and unstructured grids, and with Euler stepping, Runge-Kutta and with an implicit time integration scheme. The results from the computations have demonstrated several features. First, it has been shown that the Flux-Filter method is capable of obtaining highly accurate solution on the basis of a truly multi-dimensional approach. Second, the stability analysis has shown that the scheme does not allow too large time steps.

The integration, spatial and temporal, could be improved to allow much larger time step and convergence rates. This would be advantageous for the 3-dimensional version. One way is the use of higher order flux integration or the use of flux limiters. The temporal integration can be improved by solving the implicit method more accurately or by implementing a multigrid scheme. The improve-ments should also be aimed at reducing the level of artificial viscosity or to eleminate its use.

Extension to a 3-dimensional Flux-Filter scheme is in progress

ACKNOWLEDGEMENT

This research project was sponsered by the DFG under contract Wa424/10.

REFERENCES

[1] Steger, J. L., Warming, R.F.: Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite-Difference Methods. Journal of Computational Physics, Vol 40, pg 263-293 (1981)

[2] Van Leer, B.: Flux Vector Splitting for the Euler Equations. Proc. 8th International Conference on Numerical Methods in Fluid Dynamics, Berlin, Springer (1982)

[3] Roe, P.L.: The Use of the Riemann Problem in Finite Difference Schemes. Lecture Notes in Physics, Vol 141, pg 354-359, Berlin Springer Verlag (1981)

[4] Osher, S.: Numerical Solution of Singular Pertubation Problems and Hyperbolic Systems of Conservation Laws. Mathematical Studies, Vol 47 Amsterdam, North Holland (1981)

[5] Powell, K.G., Barth, T.J., Parpia, I.F.: A Solution Scheme for the Euler Equations Based on a Multidimensional Wave Model. AIAA 93-0065 (1993)

[6] Rossow, C.C.: Efficient Cell-Vertex Upwind Scheme for the Two Dimensional Euler Equations. AIAA Journal vol. 32, pg 278-284, (1994)

[7] Giles, M., Anderson, W., Roberts, T.: Upwind Control Volumes A New Upwind Approach. AIAA 90-0104 (1990)

[8] Vinckier, A.: An Upwind Scheme Using Flux Filters Applied to the Quasi 1-D Euler Equations. ZFW 15 (1991)

[9] Vinckier, A., Jacobsen, J., Wagner, S.: A Flux Filter Scheme Applied to the Euler and Navier Stokes Equations. Eccomas Proceedings. Stuttgart 1994. J. Wiley (1994)

[10] Lomax, H.: Finite Difference Methods for Fluid Dynamics, Lecture Notes AA214A. Stanford University (1987)

[11] Levy, D.W., Powell, K.G., Van Leer, B.: An Implementation of a Grid Independent Upwind Scheme for the Euler Equations. AIAA 89-1931 (1989)

[12] Viviand, H.: Numerical Solutions of Two-Dimensional Reference Test Cases. AGARD-AR-211, pg 6.1 (1985)

[13] Venkatakrishnan, V.: Viscous Computations Using a Direct Solver. Computers & Fluids Vol 18. No2 pg 191-204 (1990)

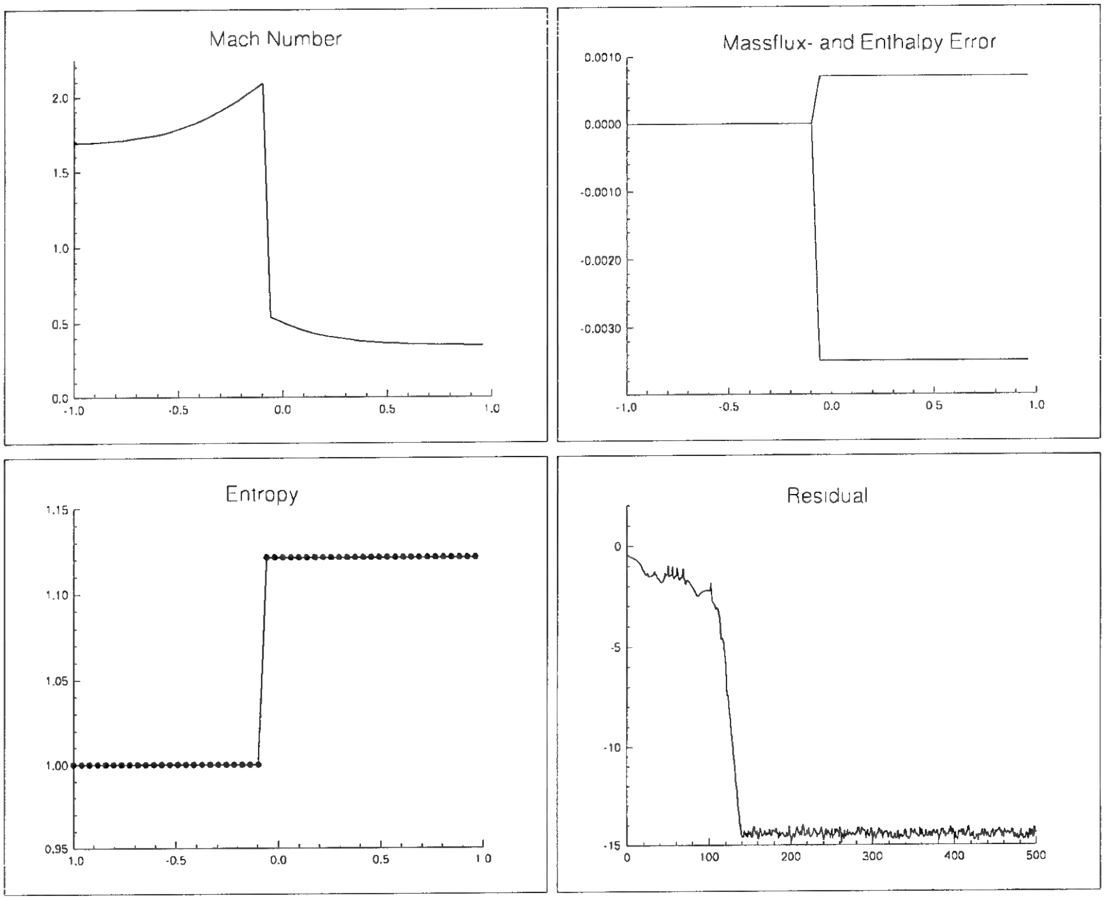


Figure 1 Solution of a supersonic-subsonic flow in a diverging nozzle.

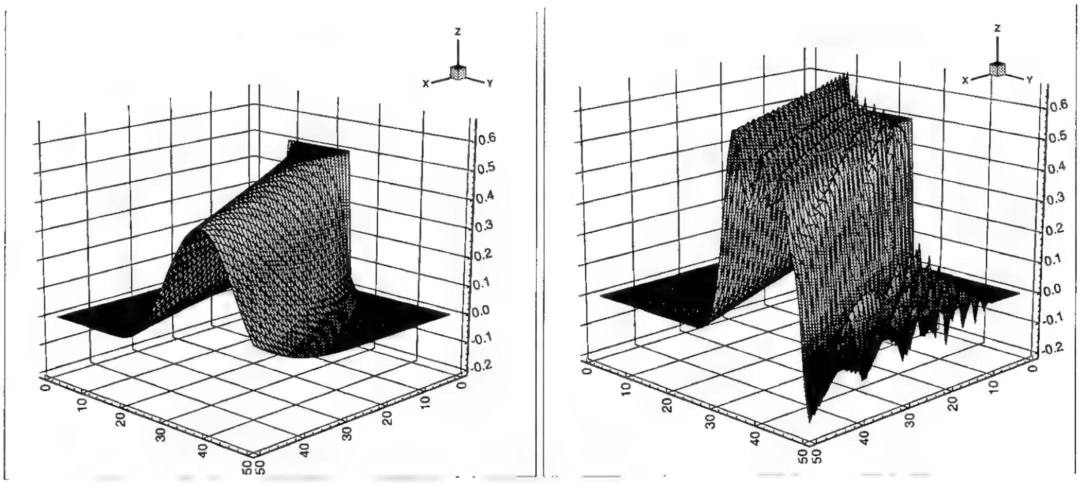


Figure 2 Solution of the scalar model equation with first order, respectively second order flux integration.

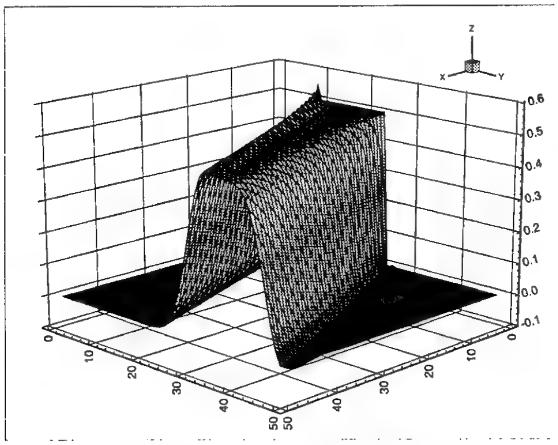


Figure 3 Solution of the scalar model equation with preferential flux integration and artificial viscosity.

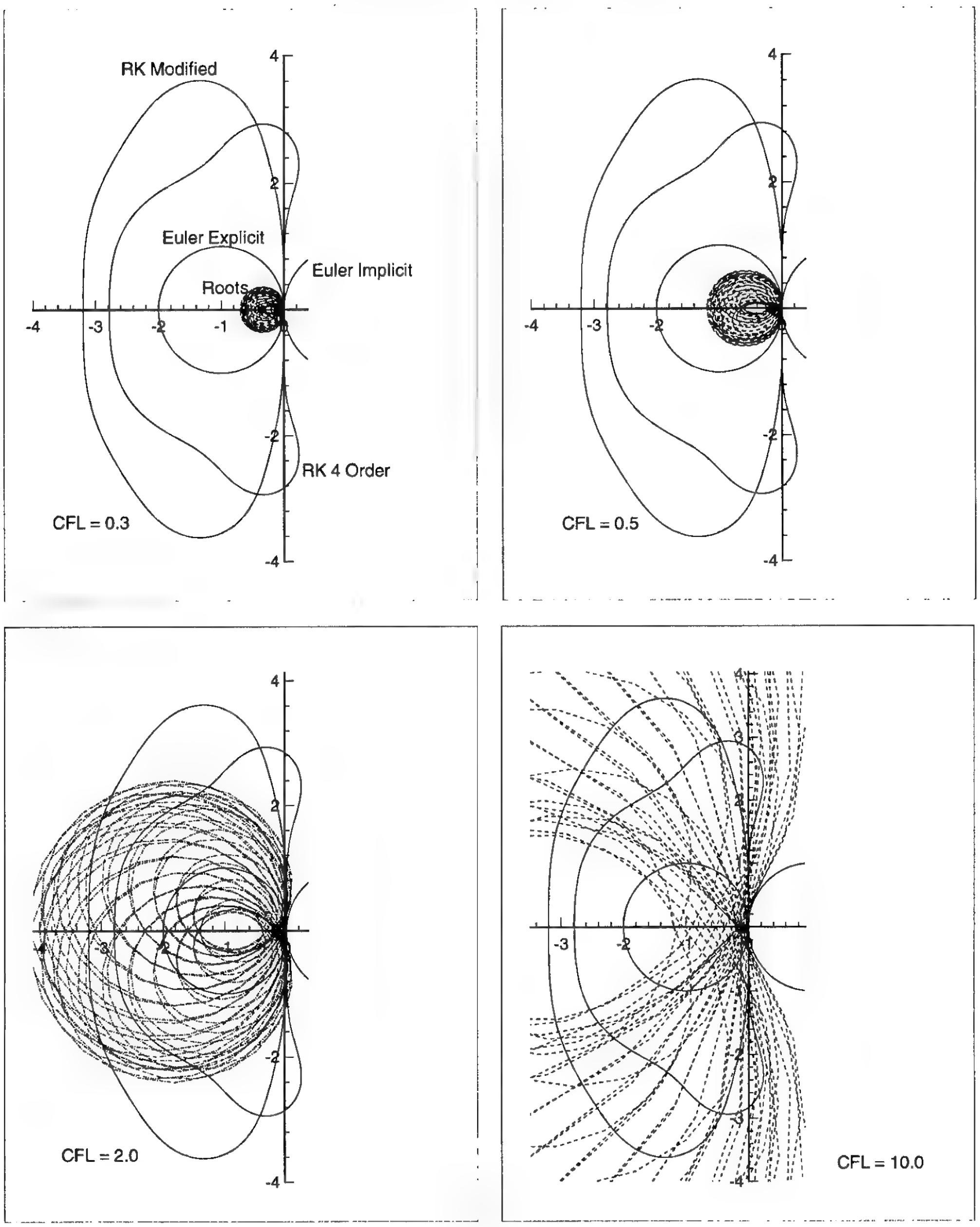


Figure 4 Spacial roots of flux filter scheme vs stability contours of different time integration methods

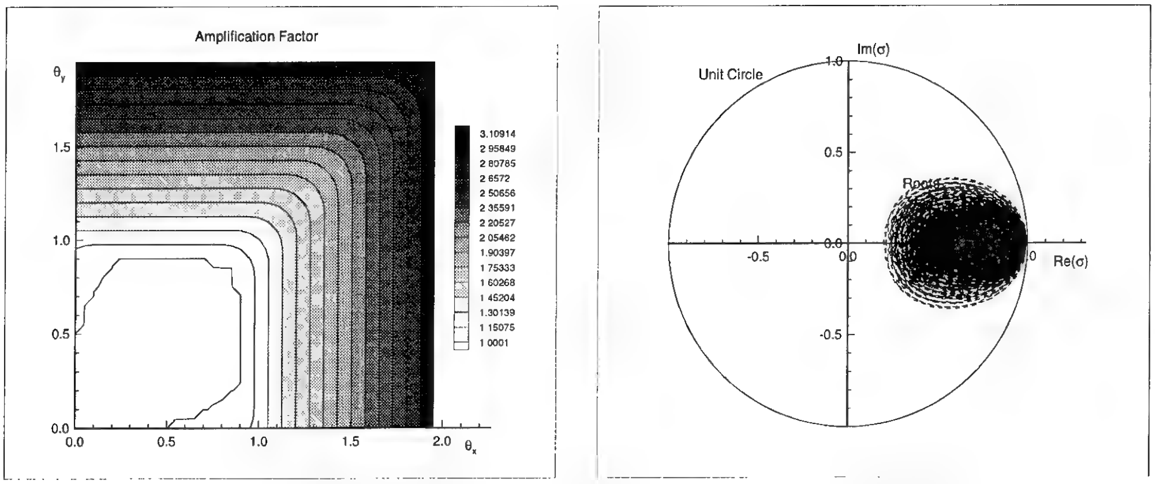


Figure 5 Amplification factor and roots (CFL = 0.3) for the flux filter scheme.

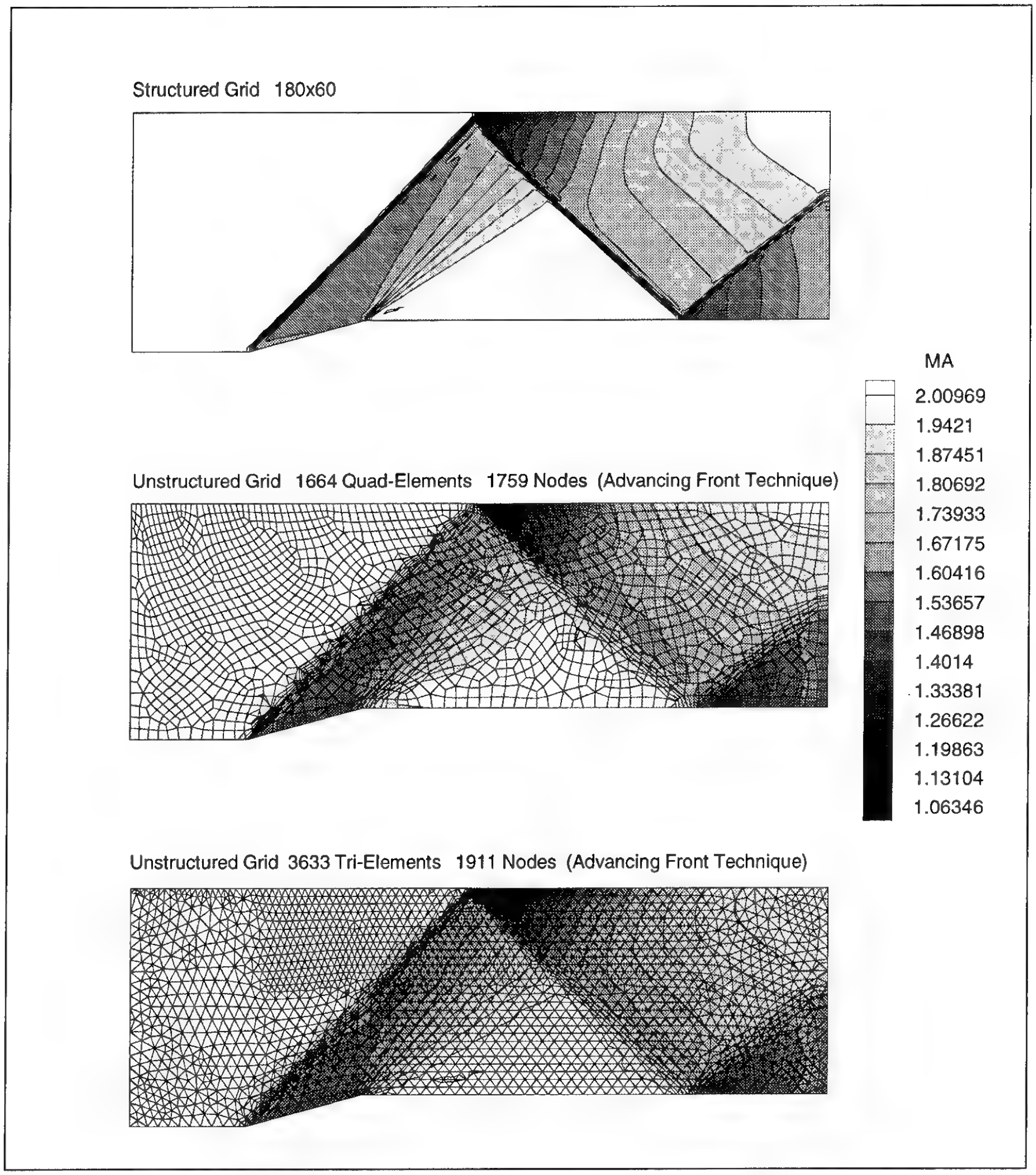


Figure 6 Supersonic flow over a 15 degree wedge solved on different grid types.

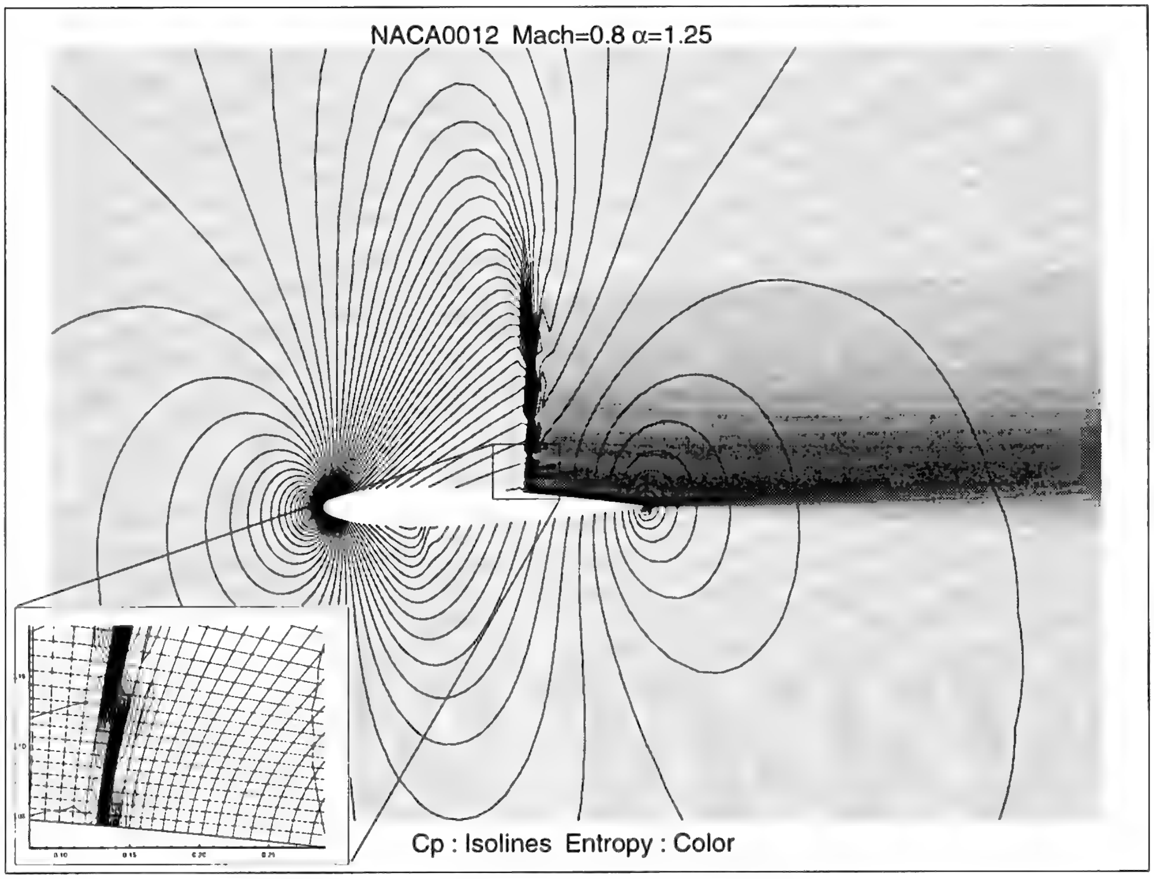


Figure 7 Transonic flow over a NACA0012 profile

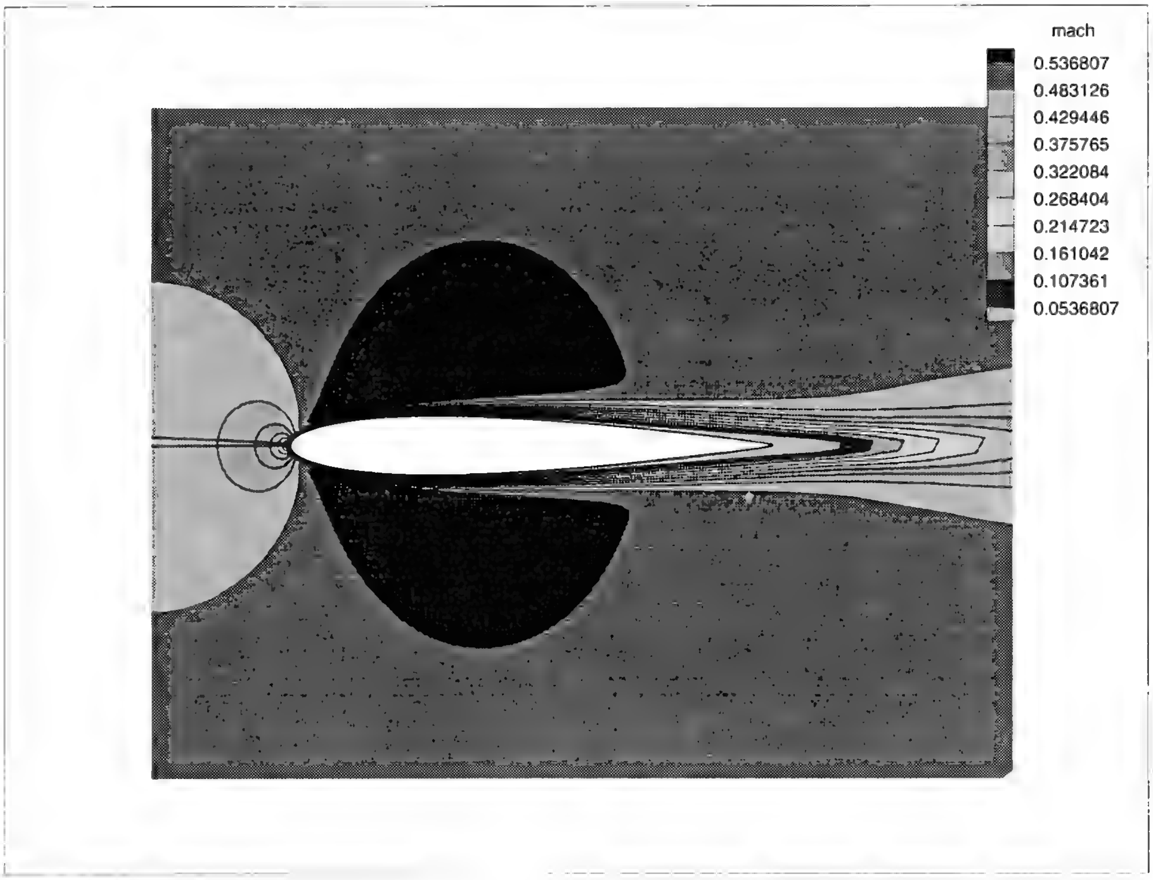


Figure 8 Navier Stokes solution around NACA0012 with a Reynolds number of 5000.

Implicit Multidimensional Upwind Residual Distribution Schemes on Adaptive Meshes

H. Paillère, J.-C. Carette, E. Issman, E. van der Weide, H. Deconinck and G. Degrez
von Karman Institute for Fluid Dynamics
Chaussée de Waterloo 72, B-1640 Rhode-St-Genèse, Belgium

Abstract

The paper reviews recent developments in multidimensional upwind schemes based on the residual decomposition or fluctuation splitting approach. Unlike the standard finite volume approach, the upwinding is based on multidimensional physics, e.g. convection of entropy and total enthalpy along the streamline and convection of acoustic Riemann invariants along the Mach lines in steady supersonic flow. The resulting schemes on triangles and quadrilaterals are very compact, with stencils consisting of nearest neighbours only and can be made monotonic and second order, like the TVD schemes in finite volumes. Numerical examples show the improved performances compared to state-of-the-art methods. The paper further describes the introduction of convergence acceleration techniques which exploit the compactness of the stencils and the implementation of solution adaptive error control. The latter is based on scalar finite element *a posteriori* error estimates which are applied to the Euler system in decoupled form thanks to the multidimensional residual decompositions.

1 THE CASE FOR MULTIDIMENSIONAL UPWINDING

At the heart of present day upwind schemes for computing compressible flows is the solution of the one-dimensional Riemann problem : it describes the evolution of the flow which results from bringing into contact two fluids at constant but different states.
Conservative Finite Volume Methods use this building block as follows : writing the conservation law for a given cell, the cell-face fluxes in the spatial operator are evaluated by solving for each face in turn, the one-dimensional Riemann problem defined by the cell averages (or a reconstruction at the cell face) on either side, thereby assuming a series of one-dimensional problems in the direction of the cell-face normals.
Although extremely successful, the question rises how truly multidimensional physics could be brought into this picture, and what benefits could be expected from doing so.
Consider therefore the case of the steady Euler equations. Choosing entropy, total enthalpy and the components of velocity as the independent variables, the equations take the following familiar quasilinear form

$$\begin{aligned} &\vec{u} \cdot \vec{\nabla} S = 0 \\ &\vec{u} \cdot \vec{\nabla} H = 0 \\ &(1 - \tfrac{u^2}{c^2})u_x - \tfrac{uv}{c^2}(u_y + v_x) + (1 - \tfrac{v^2}{c^2})v_y = 0 \\ &v_x - u_y + \alpha_1 \partial_n S + \alpha_2 \partial_n H = 0. \end{aligned} \tag{1}$$

The first two equations express that entropy S and total enthalpy H are Riemann invariants along the streamlines, which are the fieldlines of the velocity vector $\vec{u}$ with Cartesian components u and v. The third equation is the familiar compressible potential equation written in primitive variables u and v, with c the soundspeed. The fourth equation is the vorticity equation (Crocco's equation), which is coupled to entropy and total enthalpy through derivatives in the direction n normal to the streamline, $\partial_n = -\frac{v}{q}\partial_x + \frac{u}{q}\partial_y$, where $q = \sqrt{u^2 + v^2}$ is the norm of the velocity vector. The coefficients α_1 and α_2 are given by

$$\alpha_1 = -\frac{1}{(\gamma - 1)\rho q}, \qquad \alpha_2 = \frac{1}{q}, \tag{2}$$

corresponding to the definition $\partial S = \partial p - c^2 \partial \rho$. For shock free flow with uniform homentropic and homenthalpic inlet conditions, the first two equations have the trivial solution $S(x,y) = C^t$, $H(x,y) = C^t$, leading to the potential formulation for steady irrotational flow.
At this stage of the analysis, it is instructive to recall that the first two equations are ordinary differential equations, which can be integrated by marching along the streamline starting at the inlet boundary, commonly known as the method of characteristics. It is in fact remarkable that this idea of upwinding entropy and total enthalpy along the streamline is totally absent in the state-of-the-art *conservative* methods for solving the steady Euler equations. One of the key aims in multidimensional upwinding methods is precisely to re-introduce this idea in a conservative formulation. Note further that these two equations are equally valid in 3D.
Turning our attention back to the two remaining equations, the analysis simplifies by considering a streamline-aligned coordinate system with coordinates X in the streamline and Y in the normal direction. In the new axes, the velocities are denoted by $U = q$, $V = 0$, giving for the Euler equations (1):

$$\begin{aligned} &\partial_X S = 0 \\ &\partial_X H = 0 \\ &\partial_X U - \tfrac{1}{M^2-1}\partial_Y V = 0 \\ &\partial_X V - \partial_Y U + \alpha_1 \partial_Y S + \alpha_2 \partial_Y H = 0. \end{aligned} \tag{3}$$

For *supersonic flow*, the latter two equations can be expressed as ordinary differential equations along the Mach lines Γ^+ and Γ^-, (see Figure 1), by introducing the acoustic characteristic variables

$$\begin{aligned} \partial C^+ &= \alpha_1 \partial S + \alpha_2 \partial H - \partial U + \tfrac{1}{\sqrt{M^2-1}}\partial V \\ \partial C^- &= \alpha_1 \partial S + \alpha_2 \partial H - \partial U - \tfrac{1}{\sqrt{M^2-1}}\partial V \end{aligned} \tag{4}$$

In these variables the Euler system for steady supersonic flow can be written as

$$\begin{aligned} &\partial_X S = 0 \\ &\partial_X H = 0 \\ &\partial_X C^+ + \tfrac{1}{\sqrt{M^2-1}}\partial_Y C^+ = 0 \\ &\partial_X C^- - \tfrac{1}{\sqrt{M^2-1}}\partial_Y C^- = 0, \end{aligned} \tag{5}$$

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

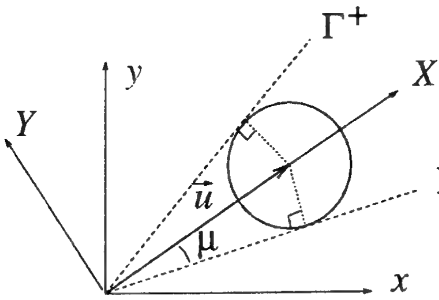


Fig. 1 : Mach angles $\pm\mu$ and streamline coordinate system (X, Y)

where $tg\mu = \frac{1}{\sqrt{M^2-1}}$ defines the Mach angle $\pm\mu$ between Machlines and streamline, see Figure 1. Again, the physics dictate a multidimensional upwinding discretization of the acoustic equations by a space operator upwinded along the *Machlines*, as in the method of characteristics. The point of deviation from the method of characteristics is to achieve this in a *conservative formulation*, such that shocks and contacts can be handled without any special treatment. This indeed is the hallmark and basis of success of the state-of-the-art Finite Volume approach. The price paid in these methods, however, is that the upwinding is based on locally one-dimensional physics, by considering the states adjacent to each finite volume face as the initial data for a one-dimensional Riemann problem in the direction of the face normal. Such an approach precludes the use of Mach lines or the streamline as the upwinding directions.

For *subsonic flow*, the two acoustic equations form an elliptic subset, and it is less clear what should be the optimal space discretization.

To fix the ideas, consider again the Euler system in the form of eq. (3), but assume that the inlet conditions are such that the flow is irrotational, so that the third and fourth equations decouple from S and H :

$$\begin{array}{l} \partial_X U - \frac{1}{M^2-1}\partial_Y V = 0 \\ \partial_X V - \partial_Y U = 0 \end{array} \qquad (6)$$

For subsonic flow, the eigenvalues of the system matrix (and hence the Mach angles) become complex $(\frac{\pm i}{\sqrt{1-M^2}})$. For $M = 0$ this is the set of Cauchy-Riemann equations governing incompressible potential flow.

Different ways for discretizing such a system will be discussed in section 2. All these methods are based on an unsteady version of eqn(5) or (6), whereby the unsteady terms are in general chosen not to be physical, but such that the resulting system is hyperbolic in time and achieves fast convergence to the steady state.

Such a choice [1] is given by the following system, both valid for subsonic and supersonic flow, and called the hyperbolic/elliptic splitting. Defining the characteristic variables

$$\partial \mathbf{W} = \begin{pmatrix} M\beta\partial C^+ \\ M\beta\partial C^- \\ \frac{\partial H}{c} - \frac{\partial S}{\rho c(\gamma-1)} \\ \partial S \end{pmatrix} \qquad (7)$$

it is given by

$$\begin{array}{l} \frac{\partial \mathbf{W}}{\partial t} + \begin{bmatrix} \chi\nu^+ & \chi\nu^- & 0 & 0 \\ \chi\nu^- & \chi\nu^+ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \frac{\partial \mathbf{W}}{\partial X} \\ \qquad + \begin{bmatrix} \frac{\chi}{\beta} & 0 & 0 & 0 \\ 0 & -\frac{\chi}{\beta} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \frac{\partial \mathbf{W}}{\partial Y} = 0 \end{array} \qquad (8)$$

or in Cartesian coordinates

$$\frac{\partial \mathbf{W}}{\partial t} + A_W \frac{\partial \mathbf{W}}{\partial x} + B_W \frac{\partial \mathbf{W}}{\partial y} = 0. \qquad (9)$$

Here ν^+ and ν^- are defined as $\nu^+ = \frac{M^2-1+\beta^2}{2\beta^2}$ and $\nu^- = \frac{M^2-1-\beta^2}{2\beta^2}$ and χ and β are given by $\beta = \sqrt{\max(\varepsilon^2, |M^2-1|)}$ and $\chi = \frac{\beta}{\max(M,1)}$. To circumvent the singularity at the sonic point, ε is different from zero and given a small value (typically 0.05).

Clearly, the third and fourth characteristic equations decouple for all flow regimes, implying as before that entropy and total enthalpy are conserved along streamlines in the steady state. Considering the first and second equations, $\nu^- = 0$ and $\nu^+ = 1$ for supersonic flow, and the equations are fully diagonal; the system is in fact identical to eqn(5), where the acoustic variables are made to propagate along the Mach lines. In the subsonic case, the system is no longer diagonal and the two acoustic equations become coupled and form a system which is elliptic at steady state.

The residual in conservative variables is obtained by transforming eqn(9), giving

$$\frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = R\left[A_W \frac{\partial \mathbf{W}}{\partial x} + B_W \frac{\partial \mathbf{W}}{\partial y}\right] \qquad (10)$$

where R is to transformation matrix from characteristic to conservative variables.

2 RESIDUAL DISTRIBUTION SPACE DISCRETIZATION

The finite volume setting with its underlying discontinuous solution representation naturally leads to the definition of 1D Riemann problems at the discontinuous interfaces, although some progress has been made in the solution of three-state two-dimensional Riemann problems [2].

Therefore, in this work, we concentrate on approaches based on a continuous representation of the solution over structured or unstructured meshes, with the solution stored at the vertices. Such a framework, which allows easy incorporation of upwinding concepts, is provided by the residual distribution approach :

- in a first step, the conservative flux balance (cell residual) is evaluated over a cell with unknowns located in the vertices by a simple contour integration (e.g trapezium or midpoint rule).
- in a second step the cell residuals are distributed to the vertices to form the nodal space operator (or nodal residual) which becomes a weighted average of the adjacent cell residuals.

The space discretization is consistent and conservative, which is easily shown by summing up the discrete equation for all nodes and observing that the interior fluxes vanish, leaving only the contributions from the boundaries.

Various residual distribution schemes have been proposed, such as the central scheme of Jameson [3] or the Lax-Wendroff schemes of Ni [4], Hall [5] and Morton [6]. In the present context the residual distribution framework has been used to formulate conservative multidimensional *upwind* advection schemes. At the scalar level, linear and non-linear advection schemes are obtained by distributing the cell residual to the *downstream* nodes only. In this way, properties such as positivity and second order accuracy (linearity preservation) [7, 8, 9] can be built-in. As long as the distributed parts sum up to the *conservative* cell residual, the schemes satisfy discrete conservation.

The application to the Euler equations, e.g. for advection of entropy, total enthalpy and acoustic variables, is straight forward, provided that a conservative linearization can be found which ensures that the flux balance over a cell can be written exactly in terms of the quasilinear equations discussed in section 2.

2.1 Distribution schemes for scalar advection

The subject of multidimensional shock-capturing advection schemes on triangles has been extensively treated in previous publications, and the reader is referred to [9, 10], as well as to the work of Roe and Sidilkover [8, 11] for details. Only the most important aspects are recalled here, and the extension to quadrilaterals is briefly discussed.

Consider the linear advection equation, with constant $\vec{\lambda}$:

$$\frac{\partial u}{\partial t} + \vec{\lambda}\cdot\vec{\nabla} u = 0 \qquad (11)$$

The corresponding integral form of eqn(11) is obtained by integrating over a control volume Ω (triangle or quadrilateral). This leads to the definition of the cell residual or fluctuation,

$$\phi^{\Omega} = -\iint_{\Omega} \frac{\partial u}{\partial t} d\Omega = \iint_{\Omega} \vec{\lambda}\cdot\vec{\nabla} u d\Omega = \oint_{\Gamma} u\,\vec{\lambda}\cdot\vec{n}_{ext}\, d\Gamma, \qquad (12)$$

where Γ is the boundary of the control volume Ω. Because the solution is stored in the vertices of the cell, the contour integral can be easily evaluated by the trapezium rule. In the fluctuation-splitting approach, fractions of ϕ^{Ω} are sent to the cell vertices, which after assembling contributions from all cells leads to the nodal update. The semi-discretization at point i is then

$$\begin{aligned} \frac{du_i}{dt} &= -\frac{1}{S_i}\sum_{\Omega}\beta_i^{\Omega}\phi^{\Omega} = -\frac{1}{S_i}\sum_{\Omega}\Psi_i^{\Omega} \qquad (13) \\ &= -R(u_i) \qquad (14) \end{aligned}$$

where S_i is the area of the median dual cell around node i, and the β_i^{Ω} are the the distribution coefficients which sum up to one for each cell. The way these coefficients are evaluated, determines the properties of the scheme. The most important of these are:

- **Positivity**
  A monotonic scheme can be obtained by demanding positivity. Suppose that the numerical solution at mesh point i is u_i. Then the positivity property requires that in the discrete form of eqn(11)

$$\frac{u_i^{n+1} - u_i^{n}}{\Delta t_i} S_i = \sum_k c_{ki}(u_k^n - u_i^n) \qquad (15)$$

the coefficients c_{ki} are all non-negative for $k \neq i$. Stability and monotonicity preservation is then guaranteed under the CFL-like condition

$$\frac{\Delta t_i}{S_i} \leq \frac{1}{\sum_k c_{ki}} \qquad (16)$$

Indeed, if u_i^n is a local maximum, i.e. $(u_k^n - u_i^n) \leq 0$ $\forall k$, then $\frac{\partial u_i}{\partial t} \leq 0$. Consequently a local maximum cannot increase and similarly a local minimum cannot decrease. The condition(15) is called global positivity and is difficult to impose for fluctuation splitting schemes. Therefore a more restrictive property is introduced, namely local positivity, see [12]. This means that condition(15) is imposed for each contributing control volume in eqn(14), which is very easy to check. Positivity will in general be linked to some upwinding. In the fluctuation splitting context upwind biasing is obtained by limiting the distribution of the cell residual to the downstream nodes.

- **Linearity Preservation or Residual Property**
  Second order truncation error in the steady state is obtained by demanding that no updates are sent to the vertices if the cell residual is zero. This is obtained when the distribution coefficients β_i^{Ω} are bounded, such that

$$\beta_i^{\Omega}\phi^{\Omega} \to 0 \qquad \text{when} \qquad \Phi^{\Omega} \to 0 \qquad (17)$$

It can be proven that only non-linear schemes (a scheme is called linear if the coefficients c_{ki} in eqn(15) are independent of u_k) can satisfy both properties.

2.1.1 Schemes on triangles

Considering the triangle with inward normals $\vec{n}_i$ shown in figure 2, the fluctuation ϕ^{Ω}, eqn(12), can be written as

$$\phi^T = \sum_{i=1}^{3} k_i u_i^n \qquad k_i = \frac{1}{2}\vec{\lambda}\cdot\vec{n}_i \qquad (18)$$

The k_i are convenient parameters in the design of upwind schemes. Since the inward normals $\vec{n}_i$ sum up to zero, one has also $\sum_i k_i = 0$. Four important distribution schemes are:

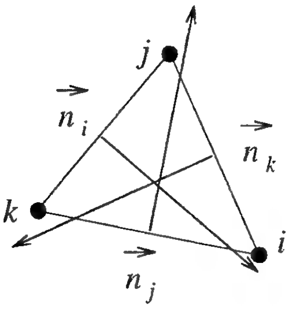


Fig. 2 : Triangle and inward normals $\vec{n}_i$

The N and PSI or limited N scheme

Define $k_i^+ = \max(0, k_i)$ and $k_i^- = \min(0, k_i)$, then the distribution to the nodes for the N scheme are given by:

$$\beta_i^N\phi^T = k_i^+(u_i - u_{in}) \qquad (19)$$

where

$$u_{in} = \left(\sum_i k_i^-\right)^{-1} \sum_i k_i^- u_i \qquad (20)$$

This scheme is positive but not linearity preserving. However, among the linear positive schemes, it has the lowest cross-diffusion. It is also the scheme with the narrowest stencil, hence the name N scheme.
From this scheme the contributions for the non-linear PSI scheme can be constructed by limiting the N scheme distribution as follows

$$\beta_i^{PSI}\phi^T = \frac{\max(0,\beta_i^N)}{\sum_j \max(0,\beta_j^N)}\phi^T. \qquad (21)$$

This is identical to Sidilkovers general limiter formula [13] when the MinMod function, $\Psi(r) = \frac{1}{2}(1 + \mathrm{sgn(r)})\min(\mathrm{r}, 1)$, is chosen. This scheme is both positive and linearity preserving.

The FV and limited FV scheme

For the first order upwind finite volume (FV) scheme on the median dual mesh the normals of the dual grid are needed, see figure 3. In terms of these normals the fluctuation is given by

$$\phi^T = k_a(u_3 - u_1) + k_b(u_3 - u_2) + k_c(u_2 - u_1) \qquad (22)$$

Depending on the signs of the dot products of the advection vector with the normals of the dual grid, ϕ^T is distributed to the nodes according to the formula

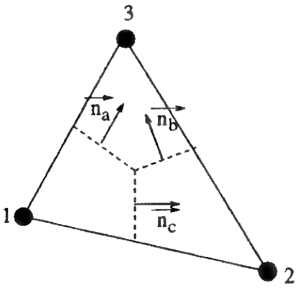


Fig. 3 : Normals for the triangle FV scheme

$$\begin{aligned} \beta_1^{FV}\phi^T &= k_a^-(u_3 - u_1) + k_c^-(u_2 - u_1) \\ \beta_2^{FV}\phi^T &= k_b^-(u_3 - u_2) + k_c^+(u_2 - u_1) \\ \beta_3^{FV}\phi^T &= k_a^+(u_3 - u_1) + k_b^+(u_3 - u_2) \end{aligned} \qquad (23)$$

Again this scheme is positive, but not linearity preserving and is more diffusive than the N-scheme. The limited second order version of this scheme is obtained by applying eqn(21) to the distribution coefficients β_i^{FV}.

The LDA scheme

For the linear LDA (Low Diffusion A) scheme, the contributions are given by:

$$\beta_i^{LDA}\phi^T = \left(\sum_j k_j^+\right)^{-1} k_i^+\phi^T \qquad (24)$$

This scheme is linearity preserving, however it is not positive.

The Lax-Wendroff scheme

The distribution coefficients for the classical Lax-Wendroff scheme are

$$\beta_i^{LW}\phi^T = \frac{1}{3} + \frac{\Delta t\ k_i}{2S_T} \qquad (25)$$

where S_T is the area of the triangle, and Δt is the Lax-Wendroff dissipation coefficient with the dimension of a time.

2.1.2 Schemes on quadrilaterals

The extension of these schemes to quadrilaterals is rather straightforward. The inner scaled normals $\vec{n}_i$ are given in figure 4. The parameters k_i are calculated as in eqn(18). The distribution coefficients of the LDA and Lax-Wendroff scheme are very similar to these on triangles and will therefore not be repeated.

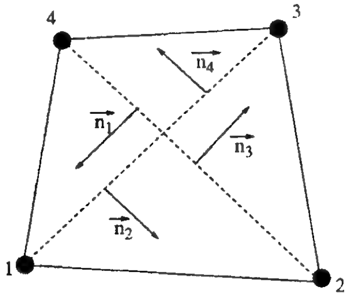


Fig. 4 : Quadrilateral and inward normals $\vec{n}_i$

The quadrilateral N and PSI scheme

The distribution coefficients for the quadrilateral N scheme are:

$$\beta_i^{NQ}\phi^Q = k_i^+(u_i - u_{i_{in}}) \qquad (26)$$

Compared with the scheme on triangles, eqn(19), every point has his own inflow state, given by

$$u_{1_{in}} = u_{3_{in}} = [-\max(|k_1|,|k_2|)]^{-1}$$

$$\left[(k_1 + |k_2|)^- u_1 + (k_3 + |k_2|)^- u_3 + k_2^- u_2 + k_4^- u_4\right]$$

$$u_{2_{in}} = u_{4_{in}} = [-\max(|k_1|,|k_2|)]^{-1}$$

$$\left[(k_2 + |k_1|)^- u_2 + (k_4 + |k_1|)^- u_4 + k_1^- u_1 + k_3^- u_3\right] \qquad (27)$$

Again this scheme is positive but not linearity preserving. The distribution coefficients for the quadrilateral PSI scheme are obtained by applying the limiter function eqn(21) to β_i^{NQ}

The quadrilateral FV and limited FV scheme

As on triangles the normals of the dual grid are needed, see figure 5. The fluctuation is

$$\begin{aligned} \phi^Q &= k_a(u_4 - u_1) + k_a(u_3 - u_2) \\ &+ k_b(u_2 - u_1) + k_b(u_3 - u_4). \end{aligned} \qquad (28)$$

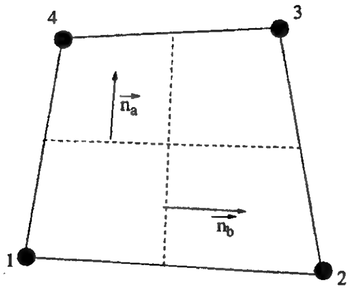


Fig. 5 : Normals for the quadrilateral FV scheme

and the distribution to the nodes

$$\begin{aligned}
\beta_1^{FVQ}\phi^Q &= k_a^-(u_4-u_1)+k_b^-(u_2-u_1)\\
\beta_2^{FVQ}\phi^Q &= k_a^-(u_3-u_2)+k_b^+(u_2-u_1)\\
\beta_3^{FVQ}\phi^Q &= k_a^+(u_3-u_2)+k_b^+(u_3-u_4)\\
\beta_4^{FVQ}\phi^Q &= k_a^+(u_4-u_1)+k_b^-(u_3-u_4)
\end{aligned} \qquad (29)$$

The limited version of this scheme is obtained as before.

2.2 System distribution schemes

As explained in section 1 the two-dimensional supersonic Euler equations can be completely decoupled and the scalar schemes of the previous chapter can be applied. However for subsonic flow the two acoustic equations, egn(8), form an elliptic subset, which cannot be decoupled. One way to treat such a system is to introduce the coupling terms as source terms and distribute them with the LDA or Lax-Wendroff scheme. By doing this the positivity property will be lost and therefore positive system distribution schemes are to be preferred. Among the system distribution schemes we mention the Lax-Wendroff and SUPG distribution. Recently, positive system distribution schemes have been explored, generalizing the scalar FV and N scheme discussed before.
Consider the unsteady hyperbolic system of equations given by

$$\frac{\partial \mathbf{W}}{\partial t}+Aw\frac{\partial \mathbf{W}}{\partial x}+Bw\frac{\partial \mathbf{W}}{\partial y}=0 \qquad (30)$$

Extending the ideas of the scalar schemes we define the matrices K_i as

$$K_i=\frac{1}{2}\left(Awn_{i,x}+Bwn_{i,y}\right) \qquad (31)$$

Because the system is hyperbolic, K_i can be written as

$$K_i=R_i\Lambda_iL_i \qquad (32)$$

where the columns of R_i contain the right eigenvectors, Λ_i is a diagonal matrix of the eigenvalues and $L_i=R_i^{-1}$. The matrices K_i^+ and K_i^- are given by

$$K_i^+=R_i\Lambda_i^+L_i, \qquad K_i^-=R_i\Lambda_i^-L_i \qquad (33)$$

Here Λ_i^+ contains the positive and Λ_i^- the negative eigenvalues.
With these definitions the system schemes on triangles can be obtained just by replacing the scalar k_i by the matrix K_i in the equations (19), (23), (24) and (25). On quadrilaterals the N-scheme in the form (26) does not generalize to systems and only system versions of the FV, LDA and Lax-Wendroff schemes can be obtained.

3 CONSERVATIVE LINEARIZATION

We now consider system (30) in conservative form

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}=0, \quad \text{or} \quad \frac{\partial \mathbf{U}}{\partial t}+\vec{\nabla}\vec{\mathbf{F}}=0 \qquad (34)$$

To maintain discrete conservation, the cell residual has to be evaluated as the flux balance of the conservative variables over the cell, for a triangle :

$$\mathbf{\Phi}^T=\oint_{\partial T}\vec{\mathbf{F}}\cdot\vec{dn}_{ext} \qquad (35)$$

On the other hand, the positive advection distribution schemes require a quasi-linear form of the residual. A conservative linearization is defined such that the quasi-linear form integrated over the surface is identical to the flux integration over the boundaries obtained by a particular integration rule. For the Euler equations on triangles, this is easily achieved by assuming that the Roe parameter vector $\mathbf{Z}=\sqrt{\rho}\,(1,u,v,H)^T$ varies linearly over each element. Since $\mathbf{U}$, $\mathbf{F}$ and $\mathbf{G}$ are quadratic in the components of $\mathbf{Z}$, the Jacobian matrices $\partial\mathbf{U}/\partial\mathbf{Z}$, $\partial\mathbf{F}/\partial\mathbf{Z}$, and $\partial\mathbf{G}/\partial\mathbf{Z}$ are linear in the components of $\mathbf{Z}$, making the integration over a triangle trivial. Defining the average state $\widehat{\mathbf{Z}}$ over the cell:

$$\widehat{\mathbf{Z}}=\frac{\mathbf{Z}_1+\mathbf{Z}_2+\mathbf{Z}_3}{3}=\frac{1}{3}\begin{pmatrix}\sqrt{\rho_1}+\sqrt{\rho_2}+\sqrt{\rho_3}\\ \sqrt{\rho_1}u_1+\sqrt{\rho_2}u_2+\sqrt{\rho_3}u_3\\ \sqrt{\rho_1}v_1+\sqrt{\rho_2}v_2+\sqrt{\rho_3}v_3\\ \sqrt{\rho_1}H_1+\sqrt{\rho_2}H_2+\sqrt{\rho_3}H_3\end{pmatrix} \qquad (36)$$

the flux balance over element T may be expressed in quasilinear form as

$$\begin{aligned}
\mathbf{\Phi}^T &= \oint_{\partial T}\mathbf{F}(\mathbf{Z})dn_y-\mathbf{G}(\mathbf{Z})dn_x && (37)\\
&= S_T\left[\widehat{A}\widehat{\mathbf{U}}_x+\widehat{B}\widehat{\mathbf{U}}_y\right] && (38)
\end{aligned}$$

where $\widehat{A}$ and $\widehat{B}$ are the analytical flux Jacobians evaluated at the average state $\widehat{\mathbf{Z}}$:

$$\widehat{A}=\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)_{\widehat{\mathbf{Z}}} \qquad \widehat{B}=\left(\frac{\partial \mathbf{G}}{\partial \mathbf{U}}\right)_{\widehat{\mathbf{Z}}}, \qquad (39)$$

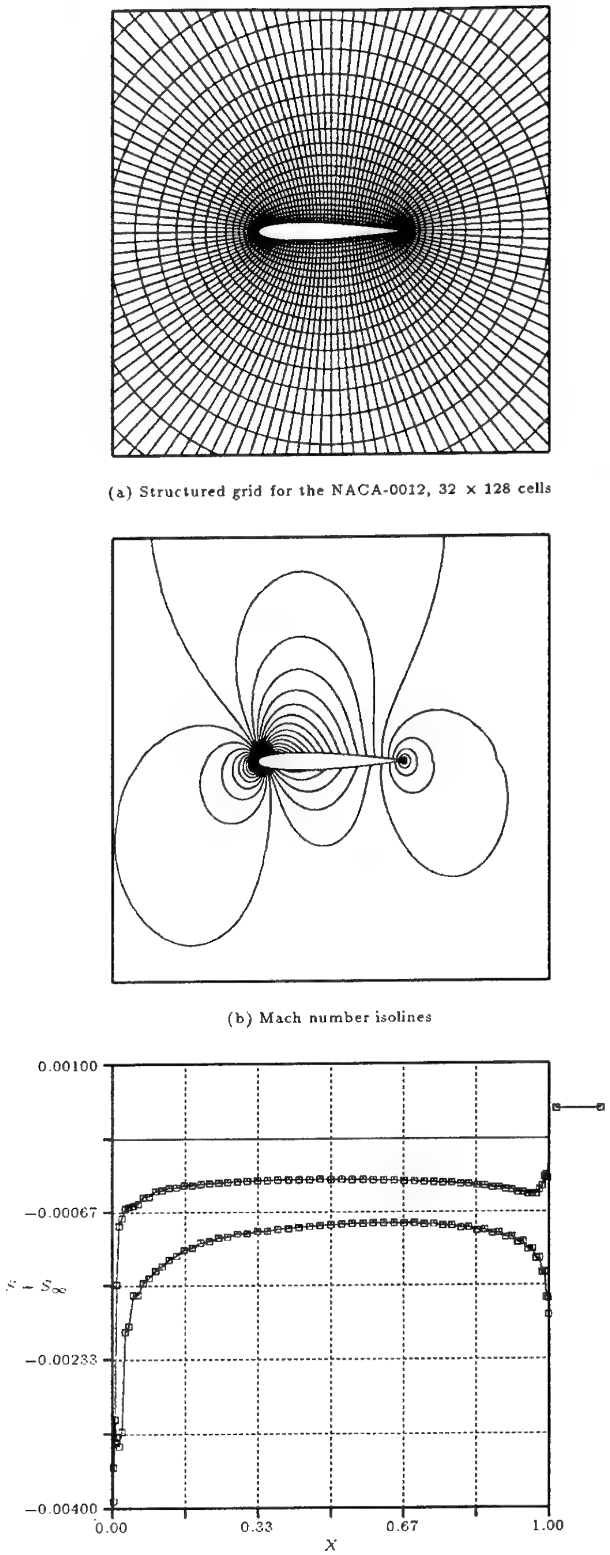
and

$$\widehat{\mathbf{U}}_x=\left(\frac{\partial \mathbf{U}}{\partial \mathbf{Z}}\right)_{\widehat{\mathbf{Z}}}\mathbf{Z}_x \qquad \widehat{\mathbf{U}}_y=\left(\frac{\partial \mathbf{U}}{\partial \mathbf{Z}}\right)_{\widehat{\mathbf{Z}}}\mathbf{Z}_y \qquad (40)$$

Because the exact Jacobians are used, one can transform (38) into any quasilinear form as long as the transformation matrices are evaluated at the average state $\widehat{\mathbf{Z}}$.
On quadrilaterals it is more difficult and for the moment a linearization is used which is only exact for parallelograms.
The global update for the system, analogous to the scalar case eqn(14), is then given by

$$\begin{aligned}
\frac{d\mathbf{U}_i}{dt} &= -\frac{1}{S_i}\sum_{\Omega}\widehat{R}D_i^{\Omega}\left[\widehat{A}_W\frac{\partial\widehat{\mathbf{W}}}{\partial x}+\widehat{B}_W\frac{\partial\widehat{\mathbf{W}}}{\partial y}\right]^{\Omega} && (41)\\
&= -\frac{1}{S_i}\sum_{\Omega}\mathbf{\Psi}_i^{\Omega}=-\mathbf{R}(\mathbf{U}_i) && (42)
\end{aligned}$$

where D_i^{Ω} is the cell distribution matrix.



(a) Structured grid for the NACA-0012, 32 × 128 cells

(b) Mach number isolines

(c) Entropy distribution on the airfoil

Fig. 6 : Mesh, Mach number isolines and Entropy distribution on the airfoil for the subsonic NACA-0012 ($M_\infty = 0.63, \alpha = 2°$), for the hyperbolic/elliptic splitting. PSI on entropy and total enthalpy, Lax-Wendroff on acoustics.

4 NUMERICAL RESULTS USING EXPLICIT TIME STEPPING

Results of three inviscid computations are given. In figure 6 the structured mesh, the Mach number isolines and the entropy distribution on the airfoil are shown for the subcritical, $M_\infty = 0.63, \alpha = 2°$, flow over a NACA-0012 airfoil. The scalar quadrilateral PSI scheme is used for the convection of entropy and total enthalpy along the streamlines, while the system Lax-Wendroff scheme is used for the coupled acoustic subsystem.

The unstructured mesh, Mach number isolines and the entropy distribution on the airfoil for the transonic NACA-0012, $M_\infty = 0.85, \alpha = 1°$, can be found in figure 7. The distribution scheme is the system PSI scheme, which allows monotonic capturing of the shock in one or two cells.

The third testcase is the thoughest, namely the hypersonic ($M_\infty = 8.7$), axisymmetric flow around a hyperboloid flare. The mesh, a triangulated structured Navier-Stokes mesh with aspect ratios over 100, and the Mach number isolines are given in figure 8. The solution is monotonic, the shock is captured very well and the carbuncle phenomenon, seen in Finite Volume solvers with Roe's approximate Riemann solver, is not present. Again the system PSI scheme was used.

5 IMPLICIT ACCELERATION

Explicit time-integration of the semi-discrete equations (42), although straightforward and robust, suffer from stability limits for some classes of problems, such as subsonic flows with stagnation regions and viscous flows. Implicit time-integration is in turn less limited by restrictions over the time-step but requires on the other hand large non-linear systems of equations to be solved.

5.1 Time-stepping strategy

As we are only interested in the steady state solution, we restrict our attention to the linearized backward Euler time-stepping scheme, which can be written as:

Loop over time: (for $k = 0, 1, \ldots$) until convergence:

- Choose time increment $\Delta^k t$,
- Compute increment Δ^k as the solution of:

$$\underbrace{\left[\frac{I}{\Delta^k t} + J_R(U^k)\right]}_{J_F(U^k)} \Delta U^k = -R(U^k), \qquad (43)$$

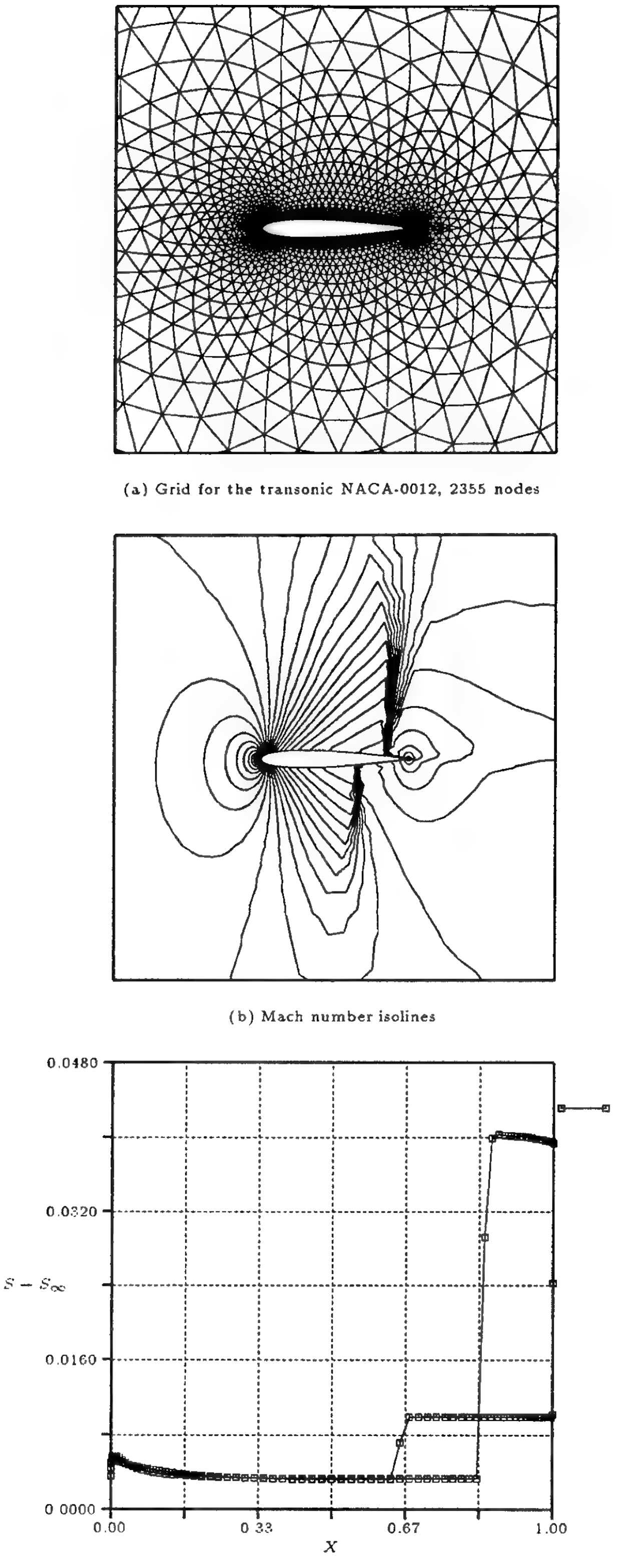
- Update: $U^{k+1} = U^k + \Delta U^k$

where $J_R(U) = \frac{\partial R(U)}{\partial U}$ is the Jacobian of the residual $R(U)$, a sparse and non-symmetric matrix, and where J_F denotes the augmented Jacobian $1/\Delta^k t + J_R$. An overview of different approaches to solve the steady state equations can be found in [14].

At each time-step k, the main ingredients of the algorithm can be listed as:

- computing a Jacobian matrix $J_R(U^k)$,
- solving the linear system (43),
- choosing a time increment $\Delta^k t$ and a non-linear update strategy

The three next subsections will be devoted to the description of each of these tasks.

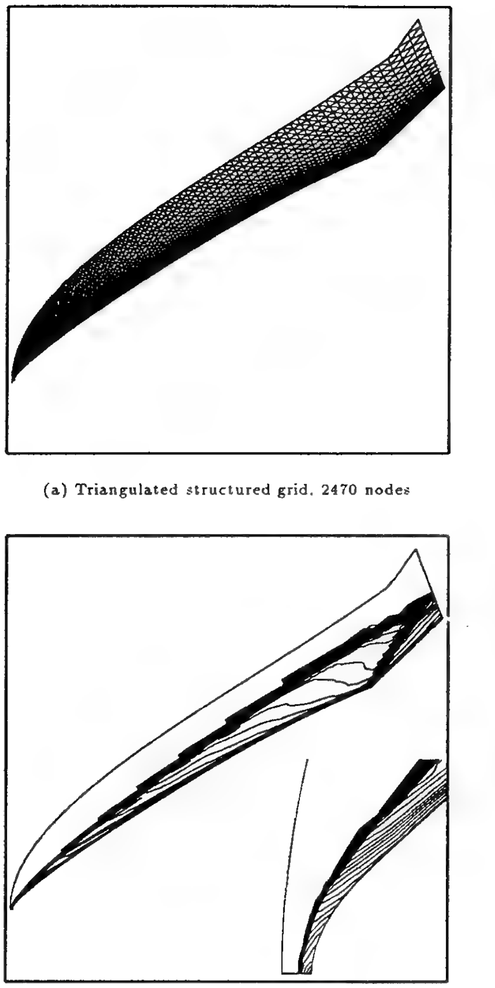


(a) Grid for the transonic NACA-0012, 2355 nodes

(b) Mach number isolines

(c) Entropy distribution on the airfoil

Fig. 7 : Mesh, Mach number isolines and Entropy distribution on the airfoil for the transonic NACA-0012 ($M_\infty = 0.85, \alpha = 1°$), for the hyperbolic/elliptic splitting. System PSI scheme.



(a) Triangulated structured grid. 2470 nodes

(b) Mach number isolines, general view and zoom of leading edge

Fig. 8 : Mesh and Mach number isolines for the hypersonic, axisymmetric hyperboloid flare, $M_\infty = 8.7$. Hyperbolic/elliptic splitting with the system PSI scheme.

5.2 Jacobian computation

5.2.1 Differentiating the Residual

As the spatial discretization stencil involves only distance-one neighbours, each individual component of the Jacobian can be computed at reasonable cost. Limiting the Taylor expansion of $\mathbf{R}_i(\mathbf{U}_j + \varepsilon \mathbf{1}_m)$, the nodal residual at node i with the m-th component of $\mathbf{U}$ at node j perturbed of a small quantity ε, to the first order terms, one has:

$$\left[\frac{\partial \mathbf{R}_i(\mathbf{U})}{\partial \mathbf{U}_j}\right]_m \simeq \frac{\mathbf{R}_i(\mathbf{U}_j + \varepsilon \mathbf{1}_m) - \mathbf{R}_i(\mathbf{U})}{\varepsilon} \tag{44}$$

It shows how each entry of the Jacobian $(J_R)_{i,j}$, a 4 × 4 matrix with m as the column index, can be computed by a first order finite difference. Because of the compactity of the scheme, this computation requires only twelve (twenty in 3D) additional explicit residual evaluations. Following the same steps of the explicit solver (i.e. loop over the cells, in each cell compute fluctuation and distribute contributions to be assembled at the nodes), the algorithm to compute the Jacobian is:

```
INITIALIZE $R(U) = 0, J_R(U) = 0$,
LOOP over triangles (T=1,2,..., nbr of cells):
  ◇ Compute fluctuation and distribute contributions
    to the 3 nodes $(i = 1,2,3) : \mathbf{R}_i \leftarrow \mathbf{R}_i + \mathbf{\Psi}_i^T$,
  ◇ LOOP over the 3 nodes of the cell (j=1,2,3):
    ◇ LOOP over the 4 components of $\mathbf{U}_j$ (m=1,2,3,4):
        - Perturb $m$-th component of $\mathbf{U}_j \leftarrow \mathbf{U}_j + \varepsilon \mathbf{1}_m$,
        - Compute new fluctuation,
        - Distribute the 3 contributions (i=1,2,3):
          $\left[\frac{\partial \mathbf{R}_i(\mathbf{U})}{\partial \mathbf{U}_j}\right]_m \leftarrow \left[\frac{\partial \mathbf{R}_i(\mathbf{U})}{\partial \mathbf{U}_j}\right]_m +$
                    $\left[\mathbf{\Psi}_i^T(\mathbf{U}_j + \varepsilon \mathbf{1}_m) - \mathbf{\Psi}_i^T(\mathbf{U})\right] / \varepsilon$
```

where $\mathbf{\Psi}_i^T(\mathbf{U}_j + \varepsilon \mathbf{1}_m)$ denotes the residual contribution to node i when the m-th component of $\mathbf{U}$ at node j has been perturbed.

A key issue to the numerical computation of the Jacobian as a finite-difference approximation is the proper choice of ε, which can be determined here on a component-by-component basis. The question is treated by Schnabel[15] who advocates:

$$\varepsilon = \sqrt{\eta} \max\left[|\mathbf{U}_{j,m}|, \mathrm{typ}(\mathbf{U}_{j,m})\right] \mathrm{sign}(\mathbf{U}_{j,m}), \qquad (45)$$

with $\mathrm{typ}(\mathbf{U}_{j,m})$ a typical user-defined order of magnitude for the m-th component of U at node j and η a lower bound on the inaccuracy in the residual $R(U)$ evaluation (relative noise). This lower bound is at best the machine-epsilon of the computer and can be larger if $R(U)$ is computed by a lengthy piece of code. Should η be worse or if $R(U)$ is not differentiable everywhere, one might rather resort to the secant method, known for multidimensional problems as the Broyden's update.

5.2.2 Broyden's method

Broyden's update method is the multidimensional extension of the secant method used for univariate problems, avoiding the need for computing any derivative. If the kth Newton-Raphson step is denoted[1] by:

$$J_R(U^k)\Delta^k U = -R(U^k),$$

with $\Delta^k U = U^{k+1} - U^k$, the generalization of the one-dimensional secant condition is that $J_R(U^{k+1})$ satisfies:

$$J_R(U^{k+1})\Delta^k U = \Delta^k R, \qquad (46)$$

where $\Delta^k R = R(U^{k+1}) - R(U^k)$. However, This does not determine $J_R(U^{k+1})$ uniquely in more than one dimension. In Broyden's update approach, $J_R(U^{k+1})$ is chosen by making the least change (see[15] for proper matrix norms) to $J_R(U^k)$, consistent with the condition (46). As such, the method suffers a major drawback as it entails a complete fill-in of the Jacobian matrix whereas the true Jacobian matrix is sparse. Alternatively, we can look for the solution to the same least change problem under the additional condition $B \in S(J_R)$ where $S(J_R)$ represents the set of $n \times n$ matrices with the same sparsity pattern as J_R. The resulting update is given by:

$$\begin{aligned} J_R(U^{k+1}) &= J_R(U^k) + \\ &\mathcal{P}_{S(J_R)}\{D^{-1}\left[\Delta^k R - J_R(U^k)\Delta^k U\right]\Delta^k U\}, \end{aligned}$$

where $\mathcal{P}_{S(J_R)}$ is the matrix operator which maps any matrix onto the same matrix but restricted to the sparsity pattern of J_R and D a $n \times n$ diagonal matrix which accounts for the sparsity structure of the Jacobian matrix:

$$D_{ii} = \delta_i^{k\,t}\delta_i^k \ \text{ with } \ (\delta_i^k)_j = \begin{cases} 0 \ \text{ if } \ (J_R)_{i,j} = 0 \\ (\Delta_i^k)_j \ \text{ otherwise} \end{cases} \qquad (47)$$

Broyden's method allows to update the Jacobian matrix without having to compute twelve residual evaluations. On the other hand, non-linear convergence will be at most linear and more iterations will be needed at the non-linear level.

5.3 Solution of the linear system

Following the linearization process, the linear system (43) is iteratively solved with left (or right) preconditioning:

$$\tilde{J}_F(U^k)^{-1}J_F(U^k)\Delta^k = -\tilde{J}_F(U^k)^{-1}R(U^k), \qquad (48)$$

with $\tilde{J}_F(U^k)$ obtained by some incomplete approximate factorization of $J_F(U^k)$. Block ILU factorization is used in our numerical experiments. Krylov subspace acceleration techniques have been considered to accelerate the convergence of the iterative solve. In the framework of this paper, we have favoured GMRES[16] among other solvers because of its optimality and since it does not represent a severe limitation for 2D medium size problems on today's computers despite its storage requirements. We refer to [17] for a description and assesment of alternate preconditioners and other Krylov subspace techniques, such as QMR and TFQMR[18]. A constant Krylov subspace dimension of 30 is used in the numerical experiments and the linear solver is stopped when the normalized linear residual drops below 10^{-5}. This linear convergence criteria is easily met within the 30 Krylov subiterations in the early stages of the convergence process when the CFL number is not too large.

5.4 Global convergence and fixed-point method

The choice of an optimal time-step is a key issue to ensure a fast and robust convergence. It seems logical to increase the time-step when approaching to the converged solution as the likelihood to be within the radius of convergence of the Newton method increases. Automatic time-increment control algorithms have been set up to relieve the user from explicitly monitoring the CFL number following the convergence level. Some experiments with such algorithms can be found in [17]. We present now a technique which consists, after some approximation, in accelerating a fixed-point method. The technique never reaches any Newton-like convergence, but shows, for a constant limited CFL number, a good global convergence behaviour. The technique consists in solving the steady-state Euler/Navier-Stokes equations with an infinite CFL number, i.e. full Newton time integration, but using a finite CFL number in the preconditioning matrix at each linearization: $J_F(U^k) = J_R(U^k)$ and $\tilde{J}_F(U^k)$ obtained as some factorization of $1/\Delta^k t + J_R$. It should be pointed out that, since the Krylov subspace dimension is not increased, this results also in solving less accurately the linear system.

The scheme, already used in [19], is building up the main features of the flow at the very early stages of the convergence process much faster than the classical backward Euler discretization in time. Asymptotically though, the method shows a monotonic linear convergence behaviour and never reaches the convergence rate, possibly quadratic, of backward Euler. The method appears therefore as complementary to backward Euler as

[1] The time-step has been eluded from the formulation. However, the argumentation which follows still holds, as backward Euler discretization in time amounts to a classical Newton's method where the increment $\Delta^k U$ has been under-relaxed for the update.

it can be used for the first non-linear iterations and provide, so doing, a well-featured initial guess for backward Euler.

The scheme can be viewed as an accelerated fixed-point method. The basic implicit technique consists in a simple relaxation procedure immediately followed by a non-linear update. The relaxation procedure is based on some approximation $\tilde{J}_F(U^k)$ of the augmented Jacobian of the residual $J_F(U^k) = 1/\Delta^k t + J_R(U^k)$, and the overall process reads:

$$\begin{aligned} U^{k+1} &= U^k - \tilde{J}_F^{-1}(U^k)R(U^k) \\ &= \mathcal{H}(U^k). \end{aligned}$$

This formulation can be seen as a particular case of the linearized backward Euler time-stepping where only one single iteration is performed at the linear level, with $\tilde{J}_F(U^k)$ as the preconditioning matrix. Then, let us define $\mathcal{G}(U) = U - \mathcal{H}(U)$ and apply Newton-GMRES to solve $\mathcal{G}(U) \equiv 0$:

$$U^{k+1} = U^k + \Delta^k \text{ with } J_G(U^k)\Delta^k = -G(U^k). \quad (49)$$

If J_G is approximated by $\tilde{J}_F^{-1} J_R$ (which is only true at the non-linear convergence), one has:

$$U^{k+1} = U^k + \Delta^k \text{ with } \tilde{J}_F^{-1} J_R \Delta^k = -\tilde{J}_F^{-1}(U^k)R(U^k),$$

which is nothing else than full Newton iterations to solve $R(U) = 0$, where the system arising at each linearization has been left-preconditioned by $\tilde{J}_F^{-1}$. In practice, the technique amounts indeed to add $1/\Delta^k t$ only in the preconditioning matrix.

Numerical experiments have shown that the accelerated fixed-point method requires CFL numbers of order $O(1).O(10)$.

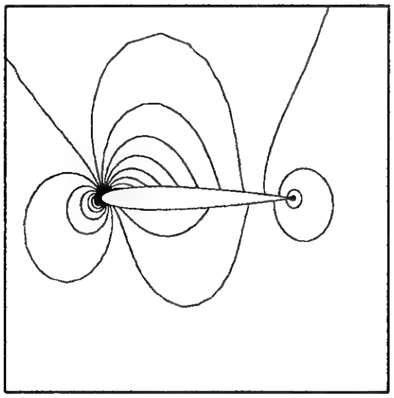


Fig. 9 : Subcritical flow over a NACA-0012: Iso-Mach contours

5.5 Numerical results

Numerical results are presented for subsonic and transonic viscous computations. Tests were performed on a DEC Alpha AXP 3000/400 workstation. The subcritical flow around a NACA-0012 airfoil at free-stream Mach number M_∞ of 0.63 and 2^o angle of attack is first considered. The grid is made of 5249 cells with far-field boundary conditions located 50 chords away from the body. Iso-Mach contours are depicted in Fig. 9. The space discretisation used the hyperbolic/elliptic splitting model and a detailed view of the grid is shown in Fig. 7. Implicit time integration was performed by updating the Jacobian with Broyden's formula, with a maximum CFL number of 200. Convergence history is shown in Fig. 10 and was achieved in about 750 CPU-seconds. In comparison, about 40000 CPU sec were needed to reduce the residual to 10^{-8} using explicit Euler time-stepping.

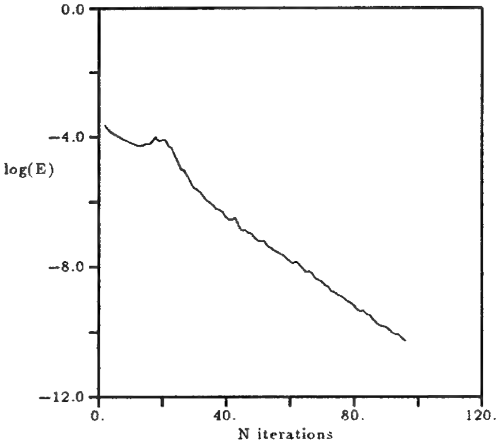


Fig. 10 : Subcritical flow over a NACA-0012: Convergence history obtained with Broyden's update, 750 CPU sec

The second test case is the viscous flow over the same airfoil with M_∞ is 2.0 and $Re = 106$, which belongs to the GAMM workshop on compressible viscous flow solver test suite ([20]). Fig. 11 shows the density contours of the solution computed with the hyperbolic/elliptic splitting model and convergence history is depicted in Fig. 12. Convergence starting from a uniform flow field with fixed point accelerated method for the two first iterations followed by backward Euler, is achieved within about 12 iterations and 350 CPU-seconds. For backward Euler, the inital CFL number of 100 was increased at every iteration by a factor $C2 = 2.0$ up to 10^6. GMRES with a Krylov subspace dimension of 50 was used for this test.

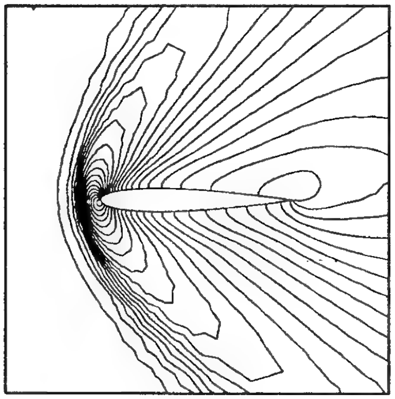


Fig. 11 : NACA-0012 $M_\infty = 2.0, Re = 106$: Density isoline contours

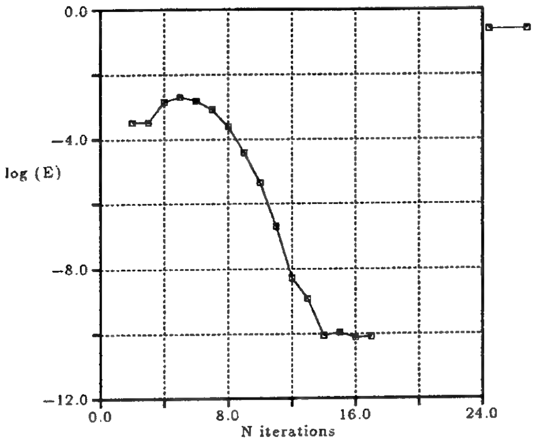


Fig. 12 : NACA-0012 $M_\infty = 2.0, Re = 106$: Convergence history, 350 CPU sec

6 MESH ADAPTIVITY

In [21], it was proposed to use the residual decomposition technique developed in the context of multidimensional upwind methods as a tool to extend the SUPG method to compressible flows. This idea was shown to lead to increased performances and robustness compared to the standard system extensions of SUPG [22, 23].
In the present section, we report the continuation of this work with focus on mesh adaptivity [24, 25] and we will show that the use of the multidimensional residual decompositions introduced to generalize the SUPG scheme to hyperbolic systems allows for the derivation of an error estimation procedure for the Euler equations in a very natural and inexpensive way.

6.1 SUPG a posteriori error estimate

The main ingredient of the proposed error estimation is the *a posteriori* error estimate developed by Johnson and Eriksson [26, 27] for the SUPG scheme applied to the following convection-diffusion equation:

$$\vec{\lambda} \cdot \vec{\nabla} u - \vec{\nabla} \cdot (\kappa \vec{\nabla} u) = f \qquad \text{in } \Omega , \qquad (50)$$

with Dirichlet boundary conditions on the boundary Γ of the computational domain Ω. If we assume that the advection vector $\vec{\lambda}$ is constant, the *a posteriori* error estimate for the scalar shock capturing SUPG scheme applied to the stationary problem (50) can be written from [27] as:

$$\|\hat{u} - U\|_{L_2(\Omega)} \leq C \| \min(1, \hat{\kappa}^{-1} h^2)\, R(U) \|_{L_2(\Omega)} + \max_{\Gamma_-} \hat{\kappa}^{1/2} \,, \qquad (51)$$

where

$$R(U) = | \vec{\lambda} \cdot \vec{\nabla} U - f \,| + \max_{S \in \partial T \subset \Omega} \left| \left[\hat{\kappa} \frac{\partial U}{\partial n_S} \right] \right| / \, h \quad \text{on } T \in \mathcal{T} \,, \qquad (52)$$

with T a triangle of mesh $\mathcal{T}$, $\hat{\kappa}$ the artificial viscosity of the SUPG scheme and n_S the normal to side S of T. Note that, for simplicity, the computed solution U is compared with the solution $\hat{u}$ of a perturbed continuous advection-diffusion problem obtained by replacing κ by $\hat{\kappa}(U)$ in eq. (50). In general, $\|u - \hat{u}\|$ is expected to be dominated by $C\|\hat{u} - U\|$, where C is a constant, so that control of $\|\hat{u} - U\|$ suffices.

6.2 Extension of the error estimation to the Euler equations

Once we are equipped with such a reliable and efficient error estimate, it is quite natural to apply this error estimate to each individual scalar equation resulting from a residual decomposition step as described in section 1. Let ϕ_k be the scalar fluctuation corresponding to equation k and ϕ_k^T the contribution of triangle T to this fluctuation. We consider then the following adaptive algorithm: Given a tolerance TOL and an initial triangulation $\mathcal{T}_0$, determine successively triangulations $\mathcal{T}_j$ with N_j elements, mesh spacings $h_j = h_j(\phi_k)$ and corresponding approximate solutions U_j, $(j = 1, \ldots, J)$, such that h_j is maximal under the local condition, for $k = 1, \ldots 4$:

$$C h_j \| \min(1, \hat{\kappa}_{j-1}^{-1} h_{j-1}^2) \frac{\phi_k^T(U_{j-1})}{S_T} \|_T \; \leq \frac{\text{TOL}}{\sqrt{N_{j-1}}}, \qquad (53)$$

on $T \in \mathcal{T}_{j-1}$ until (on the final mesh) the global norm be such that:

$$C \| \min(1, \hat{\kappa}_J^{-1} h_J^2) \frac{\phi_k^T(U_J)}{S_T} \|_{L_2(\Omega)} \leq \text{TOL} \qquad (54)$$

which is the stopping criterion. Notice that (53) seeks to equidistribute the contribution from each element to the global error bound.
From the adaptivity criterion (53), one can isolate h_j for each triangle T which provides us with a new "reference" size for each triangle. Then, it is easy to decide whether a given triangle has to be refined, coarsened or kept as it is. Of course, when dealing with the 2D Euler equations, one can compute four different required mesh sizes $h_j(\phi_k)$ for the next triangulation $\mathcal{T}_j$. At that point, several options can be taken. It could be decided for instance to control the error only on one of the 4 equations but this is risky because one could miss some of the flow features which are not "seen" by the corresponding variable. Our prefered choice therefore consists in taking the minimum of the four mesh sizes,

$$h_j^* = \min_{k=1,\ldots 4} h_j(\phi_k) \,, \qquad (55)$$

which ensures an equation-by-equation control of the error over the mesh under the required tolerance TOL.

6.3 Adaptivity technique

The adaptivity technique developed in the present research is inspired by the innovative work of Richter [28]. It consists in non-hierarchical h-refinement/derefinement allowing efficient mesh optimization operations such as edge swapping and Laplacian smoothing.
The refinement operation is achieved by the introduction of an additional node for each edge of an element for which the calculated spacing is less than the element parameter h. For interior edges, the additional node is placed at the mid-point of the edge and the solution at the new vertex is interpolated from the solution at the extremities, whereas for boundary edges, the geometrical location of the new node is determined through a spline interpolation involving the four closest existing points. For any edge that is subdivided in this manner, the two adjacent triangles associated to this edge both have to be divided in order to preserve the consistency of the final grid.
Our coarsening strategy is based on the use of a non-hierarchical data structure which enables the deletion of nodes of the initial grid and the use of the structural optimization techniques described below. The coarsening

is achieved in two steps. First, given the set of elements flagged to be deleted, a list of nodes to remove is constructed. Then, the deletion of these nodes is performed simultaneously with the reconnection of the remaining nodes to obtain a conformal mesh. This is done by identifying each element involved in the coarsening with one of the three possible derefinement cases: triangles with 1,2 or 3 nodes to be deleted (see fig. 13).

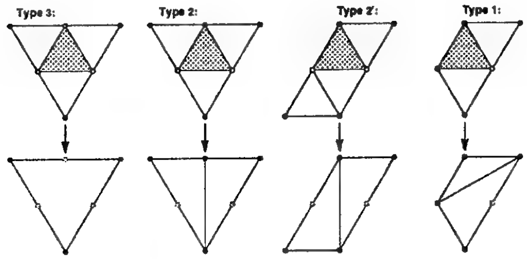


Fig. 13 : Possible coarsening configurations and their associated treatments

After the adaption step itself, a series of mesh optimization operations are performed in order to improve the quality of the grid. The first operation consists in a standard Laplacian smoothing modified in order to reduce the clustering around nodes with degree lower than 6 and the dispersion of nodes around nodes with degree higher than 6. The second operation is an edge swapping procedure which aims at minimizing the number of nodes with a high degree. This increases the number of nodes which have an optimal degree. The final operation consists in setting a minimum value to the degree of the nodes by removing undesirable low degree configurations as shown in fig. 14.

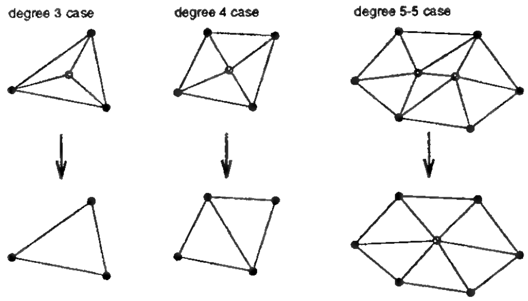
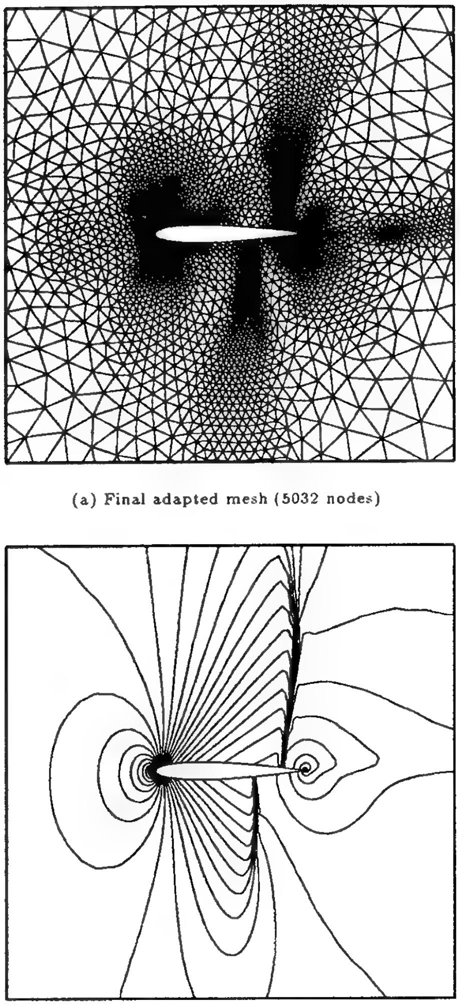


Fig. 14 : Three "pathological" low degree node configurations and their associated treatments

For more details about the adaptivity technique we refer the reader to [24].

6.4 Numerical results

The transonic flow around a NACA-0012 airfoil ($M_\infty = 0.85, \alpha = 1°$) is computed. The initial mesh is a coarse triangulation with 587 nodes and 1094 elements obtained with the frontal Delaunay method by Müller et al. [29]. The constant C appearing in (53) was chosen equal to 1, the tolerance level TOL was fixed at TOL = 0.10 and the error estimation was performed on all equations. Three adaption steps have been achieved before meeting the stopping criterion. Fig. 15 shows the final mesh as well as the Mach number isolines of the corresponding solution. The final mesh (fig. 15a) indicates clearly that all features of the flow, i.e. the stagnation zone and expansions near the leading edge, the two normal shocks and the slip line emanating from the trailing edge have been detected.



(a) Final adapted mesh (5032 nodes)

(b) Mach number isolines

Fig. 15 : Mesh adaptivity for transonic NACA-0012 ($M_\infty = 0.85, \alpha = 1°$), Scalar shock-capturing SUPG scheme associated with the hyperbolic/elliptic splitting, TOL=0.10

ACKNOWLEDGMENTS

The second and third authors are supported by a fellowship of the Belgian Fund F.R.I.A. Part of the research was supported by the CE through Bright/EuRam contract AERO-CT-0040 and the ESA MSTP program.

REFERENCES

[1] van Leer, B.; Lee, W.-T.; Roe, P.: Characteristic Time-Stepping or Local Preconditioning of the Euler equations.
1991, AIAA paper 91-1552-CP.

[2] Abgrall, R.: Approximation of the Multidimensional Riemann Problem in Compressible Fluid Mechanics by a Roe Type Method.
SIAM Journal of Numerical Analysis, 1994, Submitted for publication.

[3] Jameson, A.; Baker, T.; Weatherill, N.: Calculation of Inviscid Transonic Flow over a Complete Aircraft. 1986, AIAA-86-0103.

[4] Ni, R.-H.: A Multiple Grid Scheme for Solving the Euler Equations.
AIAA Journal, Vol. 20, 1981, pp 1565–1571.

[5] Hall, M.: Cell-vertex multigrid schemes for solution of the Euler Equations.
Numerical Methods for Fluid Dynamics, II (K.W. Morton and M.J. Baines, eds.). Oxford University Press, 1986.

[6] Morton, K.; Paisley, M.: On the cell-centre and cell-vertex approaches to the steady Euler equations and the use of shock fitting.
Lecture Notes in Physics, Vol. 264. Springer-Verlag, 1986.

[7] Roe, P.: Linear advection schemes on triangular meshes.
Cranfield Institute of Technology report, November 1987, CoA 8720.

[8] Roe, P.: "Optimum" upwind advection on a triangular mesh.
ICASE Report 90-75, 1990.

[9] Struijs, R.; Deconinck, H.; Roe, P.: Fluctuation Splitting Schemes for the 2D Euler Equations.
VKI LS 1991-01, Computational Fluid Dynamics.

[10] Deconinck, H.; Struijs, R.; Bourgois, G.; Roe, P.: Compact advection schemes on unstructured grids.
VKI LS 1993-04, Computational Fluid Dynamics, 1993.

[11] Sidilkover, D.; Roe, P.: Unification of Some Advection Schemes in Two Dimensions .
ICASE Report 95-10, 1995.

[12] Deconinck, H.; Struijs, R.; Bourgois, G.; Roe, P. L.: Compact advection schemes on unstructured grids.
Proc. of the VKI Lecture Series on Computational Fluid Dynamics. VKI LS 1993-04, 1993.

[13] Sidilkover, D.: Multidimensional upwinding and multigrid.
Proc. of the 12^{th} AIAA CFD Conference, San Diego, June 19-22, 1995. AIAA paper 95-1759-CP, 1995.

[14] Degrez, G.; Issman, E.: Solving steady compressible flow problems with subspace iteration methods.
July 3-7, 1995, 3rd Int. Congress on Industrial and Applied Mathematics, Hamburg.

[15] Schnabel, R. D.; Dennis, J. E.: Numerical methods for unconstrained optimization and non-linear equations, Prentice-Hall Series in Computational Mathematics. Prentice-Hall, 1983.

[16] Y. Saad, M. H. S.: Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems.
SIAM J. Sci. Stat. Comput., Vol. 7, No 3, July 1986, pp pp 856–869.

[17] Issman, E.; Degrez, G.; Deconinck, H.: Implicit iterative methods for a multidimensional upwind euler/navier-stokes solver on unstructured meshes.
1995. AIAA Paper 95-1653, AIAA 12th CFD Conference, San Diego, CA.

[18] R. W. Freund, N. M. N. G. H. Golub: Iterative solution of linear systems.
Acta Numerica, 1991, pp pp 57–100.

[19] Issman, E.; Degrez, G.: Convergence acceleration of a 2D Euler/Navier-Stokes solver by Krylov subspace methods.
ECCOMAS 94, 1994.

[20] Numerical simulation of compressible Navier-Stokes flows.
Bristeaux, M.-O., Glowinski, R., Periaux, J., Viviand, H. eds., Proceedings of the GAMM workshop, held at INRIA, Sophia-Antipolis (France), on December 4-6, 1985, Vol. 18 of *Notes on Numerical Fluid Mechanics*. Vieweg, 1987.

[21] Carette, J.-C.; Deconinck, H.; Paillère, H.; Roe, P. L.: Multidimensional upwinding : Its relation to finite elements.
Int. J. of Num. Meth. Fluids, Vol. 20, 1995, pp 935–955.

[22] Hughes, T. J. R.; Mallet, M.: A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems.
Comp. Meth. Appl. Mech. Engrg., Vol. 58, 1986, pp 305–328.

[23] Hansbo, P.: Explicit streamline diffusion finite element methods for the compressible Euler equations in conservation variables.
J. of Comp. Phys., Vol. 109, 1993, pp 274–288.

[24] Carette, J.-C.; Deconinck, H.: Unstructured mesh adaptivity for SUPG formulations based on residual decomposition of the euler equations.
Proc. of the VKI Lecture Series on Computational Fluid Dynamics. VKI LS 1995-02, 1995.

[25] Carette, J.-C.; Deconinck, H.: A posteriori finite element error estimation for the euler equations based on multidimensional residual decomposition.
Venice. Proc. 9^{th} Int. Conf. Finite Elements in Fluids, 1995,

[26] Johnson, C.: Adaptive finite element methods for diffusion and convection problems.
Comp. Meth. Appl. Mech. Engrg., Vol. 82, 1990, pp 301–322.

[27] Eriksson, K. E.; Johnson, C.: Adaptive streamline diffusion finite element methods for stationary convection-diffusion problems.
Math. Comp., Vol. 60, 1993, pp 167–188.

[28] Richter, R.: Schêmas de capture de discontinuités en maillage non-structuré avec adaptation dynamique. applications aux écoulements de l'aérodynamique.
PhD thesis, IMHEF, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 1993.

[29] Müller, J.-D.; Roe, P. L.; Deconinck, H.: Delaunay-based triangulations for the Navier-Stokes equations with minimum user input.
Lecture Notes in Physics, Vol. 414. Springer, 1992.

Multidimensional Upwind Dissipation for 2D/3D Euler/Navier-Stokes Applications

P. Van Ransbeeck and **Ch. Hirsch**

Department of Fluid Mechanics, Vrije Universiteit Brussel,
Pleinlaan 2, 1050 Brussels, Belgium

1. SUMMARY

Genuinely multidimensional upwind dissipation models are developed for the 2D/3D Euler/Navier-Stokes equations using a cell-centered finite-volume approach on structured grids. The numerical flux is formulated using the artificial dissipation concept. An overview is given for 2D/3D compact upwind dissipation for stencils up to respectively 6 and 8 points. A classification is set up for first and second order accurate schemes that have respectively minimum and zero cross diffusion. Second order monotone schemes are developed using the concept of non-linear limiter functions applied on multidimensional ratios of flux differences. A classification is presented for different families of 2D ratios. 3D multidimensional limiters based on 3D ratios of flux differences are introduced. The scalar dissipation models are extended and applied to the Euler/Navier-Stokes equations based on a characteristic decomposition of the inviscid operator. The resulting characteristic compatibility equations consisting of convective and source terms are depending on a set of 3 propagation directions. An overview is given for different choices of directions. The multidimensional discretisation is considered for both the convective and source terms along its associated advective speed.

2. INTRODUCTION

In the last ten years extensive research has been ongoing towards the development of genuinely multidimensional upwind schemes. The main motivation is to reduce the mesh dependency appearing in classical dimensional-split schemes and as a result to capture the physics more accurately. Two main approaches are found in literature: the fluctuation splitting schemes and the finite volume schemes, for a review see ref.[19,23,30].

The fluctuation splitting schemes consist of an upwind distribution of a fluctuation (residual) over the nodes of a triangular or tetrahedral cell[2,3,,16,17,18,21,22] In the finite volume methods the numerical flux is determined using multidimensional extrapolation[1,6,7,15,24,25]. Application of both methods to the Euler/Navier-Stokes equations consists of two basic elements : (1) a suitable wave modelling[3,17] or characteristic decomposition[8,17,18,19] of the inviscid operator and, (2) a scalar convection scheme.

The concept of artificial dissipation associated to central schemes, became a key element in Euler and Navier–Stokes calculations during the last 15 years. The family of upwind schemes, which can be considered as a rational way of defining dissipation in a numerical algorithm, has led to a matrix dissipation form, as opposed to scalar dissipation[13,29]. One of the essential elements of the upwind dissipation is the concept of non-linear limiters, leading to high resolution, 2nd order schemes, satisfying some condition of monotonicity, such as the one-dimensional concept of 'Total Variation Diminishing' TVD schemes.

Very recently a more formal approach towards a general formulation of artificial dissipation terms, applicable to structured as well as unstructured meshes is developed, based on the concept of Local Extremum Diminishing (LED) schemes, by way of generalised limiters[14]. All these developments however still remain in the dimensional splitting approach.

In this framework, 2D multidimensional upwind schemes have been reformulated as a way of defining dissipation terms, with the requirements of positivity and classical limiter concepts[9,31,33]. In contrast to the dimensional-split models, the multi-D dissipation depends on the direction of the convection speed and on variations of the solution or fluxes in different mesh directions. The corresponding numerical flux for a cell face is determined by a multidimensional interpolation inside an upstream triangle. As a result the multi-D dissipation is more compact than the models with a one-dimensional interpolation along the mesh lines. Recently a comparison and unification was performed for the underlying scalar linear and non-linear positive convection schemes for both the finite volume and fluctuation methods[11,27,32].

The idea of multidimensional limiters was first introduced for a 2D scalar convection problem[25]. Different classes of 2D limiters have been classified and applied to the 2D Euler /Navier Stokes equations[9,10,31,33]. In the present paper an overview is given for compact 2D convection schemes for stencils up to 6 points. Different classes of 2D ratios are determined by the choice of i) a triangular interpolation domain and ii) variations along meshlines or diagonals. The analysis is extended for 3D convection schemes as basis for the development of dissipation models including 3D limiters and ratios. A classification is given concerning first and second order schemes with respectively minimum and zero cross diffusion for stencils up to 8 points.

The scalar dissipation models are extended to the Euler/Navier-Stokes equations based on a characteristic decomposition[5,8] of the inviscid operator. The resulting characteristic compatibility equations represent the convection of an entropy, a shear and 2 acoustic waves. They consist of convective and source terms that depend on a set of 3 propagation directions. An overview is given of different strategies concerning the choice of the directions. The resulting equations are discretized using the scalar dissipation models. The multi-D dissipation models are considered for both the convective and source terms based on its associated advective speed.

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

3. DIMENSIONAL-SPLIT UPWIND DISSIPATION

3.1 Numerical flux formulation

Consider a 3D scalar hyperbolic conservation law based on the fluxes $\vec{f}$=(f,g,h),

$$\frac{\partial u}{\partial t} + \vec{\nabla}.\vec{f} = \frac{\partial u}{\partial t} + \vec{a}.\vec{\nabla}u = 0 \qquad (1)$$

with $\vec{a}$=(a,b,c)=(∂f/∂u,∂g/∂u,∂h/∂u) the convection speed. A cell-centered conservative finite-volume semi-discretization of (1) on a Cartesian mesh yields,

$$\frac{du_{i,j,k}}{dt} + \frac{f^*_{i+1/2,j,k} - f^*_{i-1/2,j,k}}{\Delta x} + \frac{g^*_{i,j+1/2,k} - g^*_{i,j-1/2,k}}{\Delta y} + \frac{h^*_{i,j,k+1/2} - h^*_{i,j,k-1/2}}{\Delta z} = 0 \qquad (2)$$

where e.g. the numerical flux on cell face i+1/2,j,k is expressed by

$$f^*_{i+1/2,j,k} = \frac{1}{2}(\, f_{i,j,k} + f_{i+1,j,k}\,) - d_{i+1/2,j,k} \qquad (3)$$

consisting of a central part being the average of the fluxes in the cell-centers left and right to the cell face. The numerical dissipation on cell face i+1/2,j,k is represented by $d_{i+1/2,j,k}$. All classical central and upwind dimensional-split dissipation models can be formulated as a function

$$d_{i+1/2,j,k} = d\,(\ldots,\delta u_{i-1/2,j,k},\delta u_{i+1/2,j,k},\delta u_{i+3/2,j,k},\ldots) \qquad (4)$$

depending on 1D consecutive differences of the solution along the corresponding mesh line with e.g. $\delta u_{i+1/2,j,k}=u_{i+1,j,k} - u_{i,j,k}$. For example consider the 1st and 2nd order upwind Flux Difference Splitting schemes (FDS1,FDS2) where the dissipation (4) is specified by, for a > 0,

$$d_{i+1/2,j,k} = \frac{1}{2}|a|\,\delta u_{i+1/2,j,k} - L(\delta u_{i-1/2,j,k},\delta u_{i+1/2,j,k}) \qquad (5)$$

The first term in (5) is a diffusive contribution and corresponds to first order upwinding (FDS1, L=0). Function L represents an antidiffusive term that introduces higher order accuracy,

$$L = \frac{1}{2}|a|\,\delta u_{i+1/2,j,k}\ \Phi\,(r_{i+1/2,j,k}) \qquad (6a)$$

with flux limiter Φ depending on a 1D ratio based on the sign of a as shown in figure 1,

$$\text{with} \quad r_{i+1/2,j,k} = \frac{\delta u_{i-1/2,j,k}}{\delta u_{i+1/2,j,k}} \qquad (6b)$$

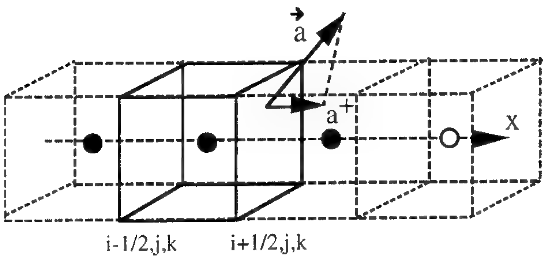


Figure 1 Second order dimensional-split upwind dissipation (a>0).

3.2 Monotonicity condition

To prevent oscillations, L is limited by use of a non-linear limiter function Φ , in order to fulfil monotonicity conditions. It assures that local maxima can not increase and local minima can not decrease. The approach in ref. [28] is used and is recently defined as Local Extremum Diminishing (LED) condition[14]. Rewriting the residual of (2),

$$\frac{du_{i,j,k}}{dt} = -\,Res_{i,j,k} = \sum_{l,m,n}^{\infty} c_{l,m,n}\,(u_{i+l,j+m,k+n} - u_{i,j,k}) \qquad (7)$$

the positivity condition is defined by

$$c_{l,m,n} \geq 0 \quad \forall \ l,m,n \qquad (8)$$

For scheme (5)-(6) this is fulfilled if the flux limiter Φ satisfies

$$0 \leq \Phi(r) \leq \min(2, 2r), \qquad \Phi\left(\frac{1}{r}\right) = \frac{\Phi(r)}{r} \qquad (9)$$

Conditions (9) are valid for all classical TVD limiters. In 1D the monotonicity concept is equivalent with the Total Variation Diminishing (TVD) approach.

4. MULTIDIMENSIONAL UPWIND DISSIPATION

4.1 2D Scalar Upwind Dissipation

In the following an overview is given for compact 2D scalar upwind schemes, including linear and non-linear classes having first and second order monotone schemes. This study is based on a theoretical analysis of 2D linear convection schemes of which the basic elements are in ref.[6] It is a generalisation of the analysis of ref.[24] where only first order optimum schemes are considered. This general study is set up for cell-centered molecules with a finite volume and structured approach. It is based on a general 9-point stencil that is derived in cartesian and streamline coordinates. Conditions concerning second order accuracy, cross diffusion, monotonicity and relations between some of these are investigated.

In contrast to the classical dimensional-split dissipation, the multidimensional upwind dissipation is based on variations of the solution in different mesh directions and on the total convection speed (a,b). The domain of dependence of the multi-D upwind dissipation models is taken in an upstream direction to the convection speed.

In the following, the linear form of (1) is considered on a uniform mesh with mesh spacing Δx=Δy=1. The fluxes are f=au and g=bu with constant convection speeds a,b>0. A general form of the multi-D upwind dissipation on face i+1/2,j, can be written as

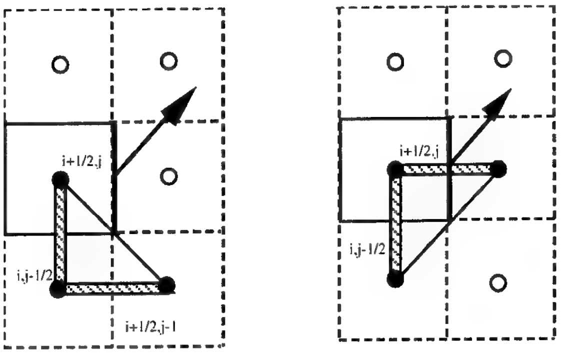
$$d_{i+1/2,j} = \frac{1}{2}|a|\,\delta u_{i+1/2,j} + L\,(\,\delta u_x, \delta u_y\,) \qquad (10)$$

The first term is the classical first order upwind dissipation from equation (5). The second term L represents the multi-D dissipation that is function of differences of the solution in both mesh directions.

Two options have been investigated for the choice (δu_x,δu_y) in (10) leading to compact 2D upwind schemes,and are illustrated in figure 2. Both variations in x and y- direction determine a triangular domain of dependence for the multi-D artificial dissipation.

4.1.1 *Linear 6-point upwind schemes*

For the linear subclass of (10) the numerical flux is determined by a linear interpolation in the corresponding triangles of figure 2, e.g. for configuration (I),



(a) config. (I)

(b) config. (II)

Figure 2 Interpolation domain I and II for the scalar multi-D upwind dissipation (a,b>0)

$$f^{*(I)}_{i+1/2,j} = au_{i,j} - \alpha\,\delta u_{i,j-1/2} + \beta\,\delta u_{i+1/2,j-1} \qquad (11)$$

Using the definition of the numerical flux (3), the multi-D upwind dissipation model is determined from (11),

$$d^{(I)}_{i+1/2,j} = \frac{1}{2}|a|\delta u_{i+1/2,j} + \alpha\,\delta u_{i,j-1/2} - \beta\,\delta u_{i+1/2,j-1} \qquad (12)$$

with positive interpolation coefficients α and β depending on a and b. Similar formulas are valid for the fluxes on cell faces i,j±1/2 introducing analogue coefficients δ and γ.

Interesting to notice is the sign of the multi-D contributions in (12). The term based on coefficient β and defined in the same direction as the 1st order term reduces the dissipation as for classical higher order schemes. While the term depending on α in the other mesh direction increases the dissipation (12). This addition of dissipation is not a loss of accuracy, on the contrary it reduces the diffusion in the cross flow direction as shown in ref.[10].

Writing out the residual, the resulting 6-point families are determined by 3 parameters A=α+δ, β and γ. Figure 3 shows for both configurations the interpolation triangles for the four cell faces.

Configuration (I) (figure 3a) consists of only 3 triangles because the interpolation domain for cell faces i-1/2,j and i,j-1/2 are identical. Configuration (II) in figure 3b consists of 4 triangles where the continuously shaded areas are referring to cell faces i±1/2,j.

Both families of schemes (figure 3) have subclasses of 2nd order accurate schemes, illustrated in figure 4. The subclass of **zero cross diffusion** schemes or second order accurate schemes for the homogeneous convection equation, is a two parameter family for both options I and II. A comparative study performed between the fluctuation splitting unstructured multidimensional upwind schemes of Deconinck and co-workers, ref.[3] and the present dissipation models is performed in ref.[32] It shows that the Low Diffusion schemes A and B (LDA and LDB) are 5-point zero cross diffusion schemes of the 6-point family of configuration (II) (figure 3b).

The 5-point continuous interpolation scheme (config. I) mentioned in figure 4 is investigated in ref.[10,15,33] and is based on a continuous interpolation for the numerical flux inside the polygon formed by the 6 surrounding cell centers of the cell-face.

A more severe constraint is the condition for general **second order accuracy**, in the classical sense, defining a unique member of the class of compact zero cross diffusion second order schemes for the non-homogeneous convection equation. Notice that for the configuration I the scheme is an upwind scheme while for configuration II the scheme is the classical central scheme

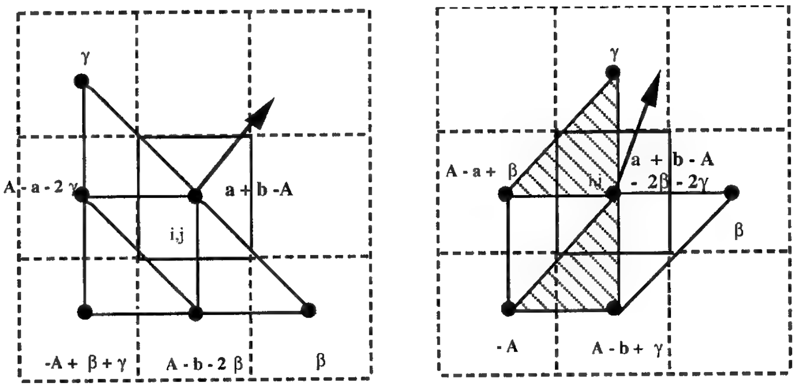
4.1.2 *Linear 4-point upwind schemes*

Both 6-point families have a subclass of 4-point stencils in common with the choice of β=γ=0 in (12) and figure 3, yielding the numerical dissipation for a,b>0

$$d_{i+1/2,j} = \frac{1}{2}|a|\,\delta u_{i+1/2,j} + L\,(\,\delta u_{i,j-1/2}\,) \qquad (13)$$

with $$L\,(\,\delta u_{i,j-1/2}\,) = \alpha\,\delta u_{i,j-1/2}$$

The resulting 4-point schemes are actually a one-parameter (A=α+δ) family of schemes, although the parameters α and δ can be chosen independently. The general 4-point scheme is splitted in a **central** part and **dissipation** term:



(a) config. (I)

(b) config. (II)

Figure 3 Six-point linear schemes, config. I and II

$$\mathrm{Res}_{i,j} = \frac{1}{2}\left(a\,\bar{\delta}_x + b\,\bar{\delta}_y \right)u_{i,j} - D_{i,j} \tag{14}$$

where the **upwind dissipation** is formulated as

$$D_{i\,j} \;=\; \frac{1}{2}\left(a\,\delta_x^+\delta_x^- + b\,\delta_y^+\delta_y^- + 2\,A\,\delta_x^-\,\delta_y^- \right)u_{i\,j} \tag{15}$$

by use of $\bar{\delta}$, δ^+ and δ^- that represent respectively central, forward and backward differences. The first 2 terms in (15) represent the 1st order dimensional-split upwind dissipation. The additional mixed 2nd difference term represents the multidimensional dissipation. The parameter **A**, determines the amount of **multidimensional upwind dissipation**.

Several interesting schemes are recovered by choosing a particular value of A as illustrated in figure 5. Concerning the subclass of 4-point **monotone first order** schemes, the lower limit (A=0) corresponds to the 1st order classical upwind scheme that has maximum cross diffusion (FDS1). The upper limit (A=min(a,b)) corresponds to the **minimum cross diffusion** scheme. This scheme has been investigated in different formulations, ref.[2,7,20,24,25]. The 4-point family has a unique non monotone **zero cross-diffusion** scheme being second order accurate for the homogeneous convection equation, ref.[6,25]. Since this scheme is a linear second order scheme it can not be monotone and shows oscillations near discontinuities.

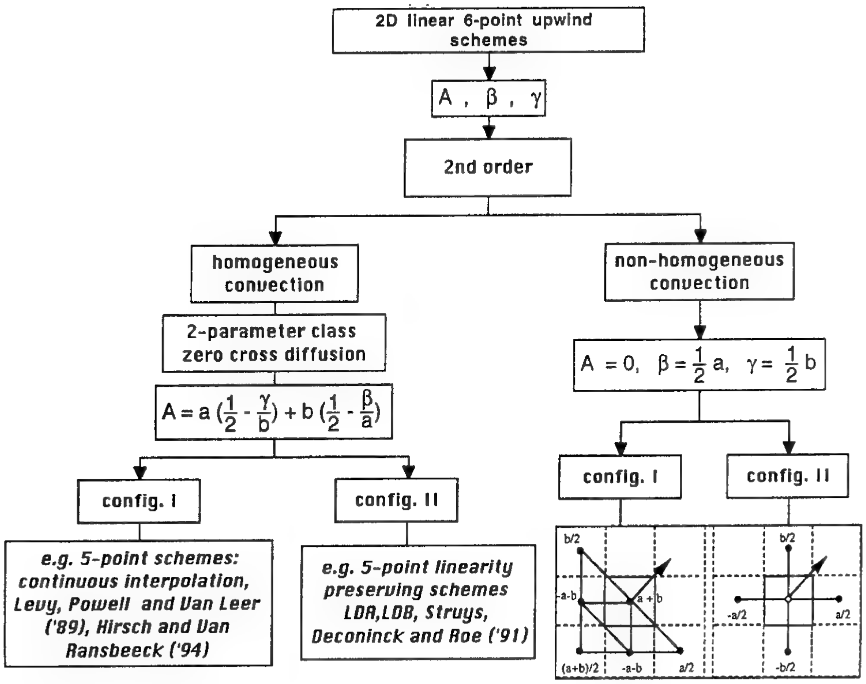

2D linear 6-point upwind schemes
A , β , γ
2nd order
homogeneous convection
non-homogeneous convection
2-parameter class zero cross diffusion
A = a (1/2 - γ/b) + b (1/2 - β/a)
A = 0, β = 1/2 a, γ = 1/2 b
config. I
config. II
config. I
config. II
e.g. 5-point schemes: continuous interpolation, Levy, Powell and Van Leer ('89), Hirsch and Van Ransbeeck ('94)
e.g. 5-point linearity preserving schemes LDA,LDB, Strugs, Deconinck and Roe ('91)
b/2
-a-b
a+b
(a+b)/2
-a-b
a/2
b/2
-a/2
a/2
-b/2

Figure 4 Overview 2nd order 6-point upwind schemes

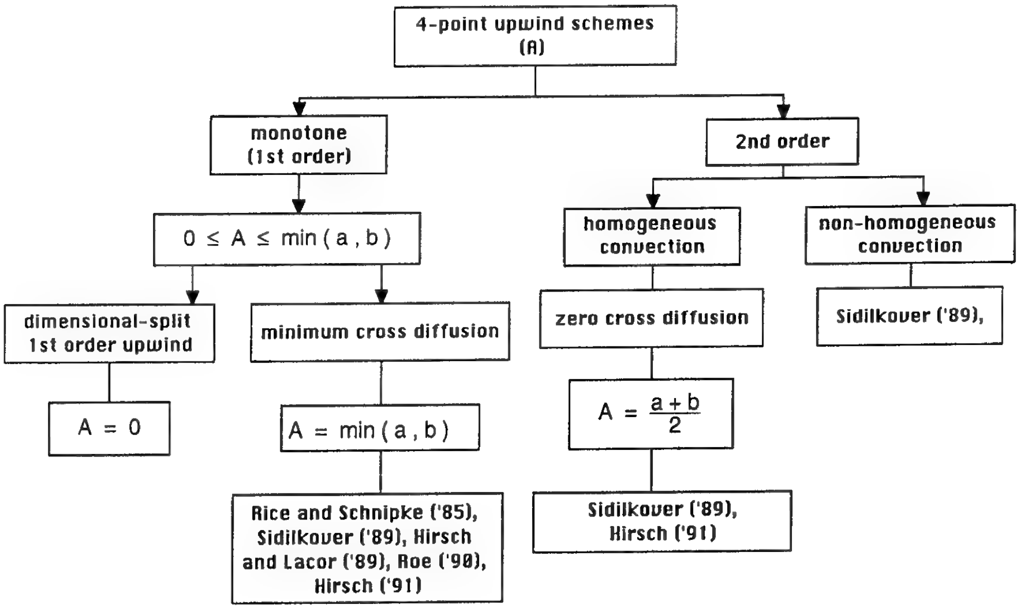

4-point upwind schemes (A)
monotone (1st order)
2nd order
0 ≤ A ≤ min (a , b)
homogeneous convection
non-homogeneous convection
dimensional-split 1st order upwind
minimum cross diffusion
zero cross diffusion
Sidilkover ('89),
A = 0
A = min (a , b)
A = (a + b)/2
Rice and Schnipke ('85), Sidilkover ('89), Hirsch and Lacor ('89), Roe ('90), Hirsch ('91)
Sidilkover ('89), Hirsch ('91)

Figure 5 2D 4-point upwind schemes

For more details about the 2D convection schemes and dissipation including 4-point,5-point and 6-point schemes, see e.g. ref.[10,11].

4.1.3 *2D Multidimensional Limiters*

The idea of multidimensional limiters was first introduced by Sidilkover, ref. [25] in the framework of a scalar convection problem. Hirsch and Van Ransbeeck, ref.[9,10,11,31] considered various multidimensional dissipation formulations based on positivity and classical limiters. To illustrate the methodology the unique 4-point zero cross diffusion scheme (fig.5) is considered below.

The definition of multidimensional limiters follows the 1D methodology. Starting with a 1st order monotone scheme, limited antidiffusive terms are added. One of the main differences with the dimensional-split limiters is that as 1st order scheme the minimum cross diffusion scheme from fig.5 is selected, having a higher accuracy than the classical 1st order scheme, e.g. ref.[31]. The 2nd order dissipation is rewritten as the first order dissipation plus anti-diffusive limited correction term,

$$d^{(2)}_{i+1/2,j} = d^{(1)}_{i+1/2,j} + L(\delta u_x , \delta u_y) \qquad (16)$$

with

$$L_{i+1/2,j} = \Delta\alpha\, \delta u_{i,j-1/2}\, \Phi (r_{i+1/2,j})$$

$$\Delta\alpha = \alpha^{(2)} - \alpha^{(1)} = \frac{1}{2} (b - \min(a,b))$$

where $\Delta\alpha$ represents the difference in interpolation coefficient between 2nd and 1st order scheme. Near discontinuities the limiter is switched off (Φ=0) and the 1st order multi-D dissipation is applied. In smooth regions Φ =1 and then the linear 2nd order scheme is applied.

The definition of the multi-D ratio in (16) and the corresponding variations δu_X and δu_Y are related to the choice of triangular interpolation domains I or II from figure 2. The following definition of general 2D ratio is used,

$$r_{i+1/2,j} = \frac{c_1\, \delta u_x + c_2\, \delta u_y}{\delta u_y} \qquad (17)$$

and is related to the choice of a triangular interpolation. Two triangle configurations are shown in figure 6. For each configuration two options are considered when fixing δu_Y and with the numerator of (17) taken as: a variation along x-direction or a variation along the diagonal.

In ref. [10] monotonicity conditions on the coefficients in (17) are derived for the 4 classes. All 2D ratios found in literature e.g.[25,26,27,32] fit in one of these classes.

4.2 3D Scalar Upwind Dissipation

In the following a brief overview is given of a theoretical analysis of 3D linear convection schemes of which the basic elements are in ref. [32] and which will be discussed in more details elsewhere. This study is based on the extension of the 2D analysis discussed in section 4.1. A general form of the multi-D upwind dissipation on face i+1/2,j,k, similar to (10) can be written as

$$d_{i+1/2,j,k} = \frac{1}{2}|a|\, \delta u_{i+1/2,j,k} + L (\delta u_x, \delta u_y, \delta u_z) \qquad (18)$$

The first term is the classical first order upwind dissipation from equation (5). The second term L represents the multi-D dissipation that is function of differences of the solution in the three mesh directions.

4.2.1 *Linear 8-point upwind schemes*

In the following, the linear form of (1) is considered on a uniform mesh with mesh spacing Δx=Δy=Δz=1. The fluxes are f=au, g=bu and h=cu with constant convection speeds a,b,c>0. The 8-point molecules, for a,b,c>0 are defined by the following extrapolation formula on e.g. face i+1/2,j,k,

$$f^{*}_{i+1/2,j,k} = au_0 - \alpha_x(u_0 - u_2) - \beta_x(u_0 - u_4) - \gamma_x(u_2 - u_6) \qquad (19)$$

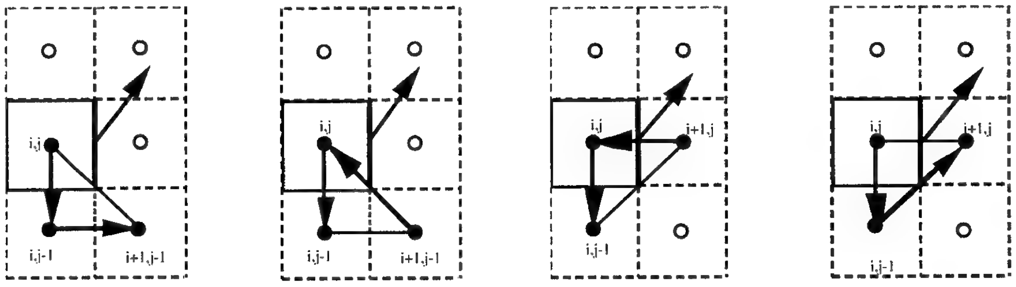
referring to figure 7 for the overall configuration of the scheme. Using the definition of the numerical flux (3), the multi-D upwind dissipation model (18) is determined from (19),

$$d_{i+1/2,j\,k} = \frac{1}{2}|a|\, \delta u_{i+1/2,j\,k} + L (\delta u_{i,j-1/2,k}, \delta u_{i,j,k-1/2}, \delta u_{i,j-1,k-1/2}) \qquad (20)$$

with

$$L = \alpha_x \delta u_{i,j-1/2,k} + \beta_x \delta u_{i,j,k-1/2} + \gamma_x \delta u_{i,j-1,k-1/2}$$

with interpolation coefficients α_X, β_X and γ_X depending on a, b and c. Similar formulas are valid for the fluxes on cell faces i,j±1/2,k and i,j,k±1/2 using respectively the sets (α_y, β_y,γ_y) and (α_Z, β_Z,γ_Z).



Ia (a) Ib (b) IIa (c) IIb (d)

Figure 6 Four classes of 2D multi-d ratios

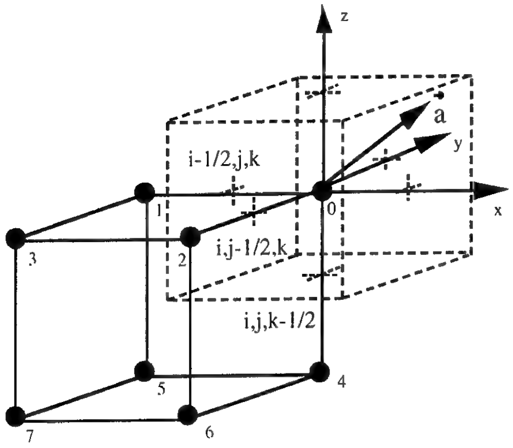


Figure 7 3D upwind schemes for 8-point stencils

It is important to observe that the resulting molecule actually represents a **four-parameter** family of schemes when the following set of parameters is chosen

$$
\begin{aligned}
A &= \alpha_x + \alpha_y + \gamma_y \\
B &= \beta_y + \beta_z \\
C &= \beta_x + \gamma_x + \alpha_z + \gamma_z \\
D &= \gamma_x + \gamma_y + \gamma_z
\end{aligned}
\qquad (21)
$$

Based on (19)-(21) the following general 8-point scheme is recovered which is splitted in a **central** part and **dissipation** term:

$$\mathrm{Res}_{i,j,k} = \frac{1}{2}\left(a\,\bar{\delta}_x + b\,\bar{\delta}_y + c\,\bar{\delta}_z\right)u_{i,j,k} - D_{i,j,k} \qquad (22)$$

where the **upwind dissipation** is formulated as

$$
\begin{aligned}
D_{ijk} &= \frac{1}{2}\left(a\,\delta_x^+\delta_x^- + b\,\delta_y^+\delta_y^- + c\,\delta_z^+\delta_z^-\right)u_{ijk} \\
&+ \left(\mathbf{A}\,\delta_x^-\delta_y^- + \mathbf{B}\,\delta_y^-\delta_z^- + \mathbf{C}\,\delta_z^-\delta_x^- - \mathbf{D}\,\delta_x^-\delta_y^-\delta_z^-\right)u_{i,j,k}
\end{aligned}
\qquad (23)
$$

The first 3 terms in (23) represent the 1st order upwind dissipation. The additional terms represent the multidimensional dissipation which consists of mixed 2nd and 3rd differences. The parameters **A,B,C** and **D** determine the amount of **multidimensional upwind dissipation**. Choosing a specific scheme corresponds to fixing the 4 parameters A,B,C and D. Since the cell face values are determined by 9 interpolation coefficients, every scheme has 5 degrees of freedom in choosing the interpolation coefficients in (21).

4.2.2 *First order monotone schemes*

Figure 8 shows a classification for the 8-point family of upwind schemes including monotonicity and zero cross diffusion conditions. The lower limit of the monotonicity condition corresponds with the classical first order upwind scheme with a maximum amount of cross diffusion. The upper limit corresponds to the minimum cross diffusion scheme also identified in ref.[24]. In the 2D case (e.g. c=0) this scheme reduces to the 2D minimum cross diffusion scheme of fig. 5 investigated before in e.g. ref.[6,24,25]. Different interpolation strategies are investigated in ref. [32].

4.2.3 *Subclass of zero cross diffusion schemes*

Evaluating the zero cross diffusion condition in fig.8 one can notice that there is no condition on parameter D. As a result there is a one-parameter subclass of zero cross diffusion schemes. Different interpolation strategies in (20) lead to different values of D. The arithmetic average procedure corresponds with the scheme used by Roe and Sidilkover as starting point in their theoretical analysis for first order

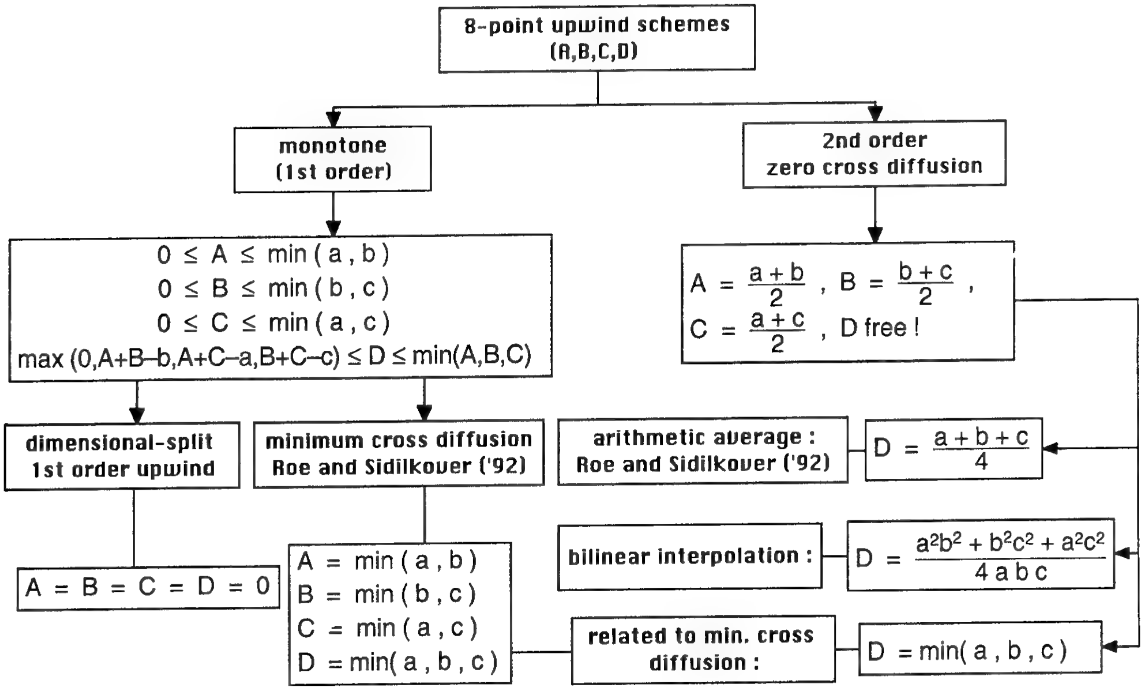

8-point upwind schemes (A,B,C,D)
monotone (1st order)
2nd order zero cross diffusion
$0 \le A \le \min(a,b)$
$0 \le B \le \min(b,c)$
$0 \le C \le \min(a,c)$
$\max(0, A+B-b, A+C-a, B+C-c) \le D \le \min(A,B,C)$
$A = \frac{a+b}{2}$, $B = \frac{b+c}{2}$,
$C = \frac{a+c}{2}$, D free !
dimensional-split 1st order upwind
minimum cross diffusion Roe and Sidilkover ('92)
arithmetic average : Roe and Sidilkover ('92)
$D = \frac{a+b+c}{4}$
$A = B = C = D = 0$
$A = \min(a,b)$
$B = \min(b,c)$
$C = \min(a,c)$
$D = \min(a,b,c)$
bilinear interpolation :
$D = \frac{a^2b^2 + b^2c^2 + a^2c^2}{4abc}$
related to min. cross diffusion :
$D = \min(a,b,c)$

Figure 8 Classification of 3D upwind schemes for 8-point stencils

optimum linear schemes in two and three dimensions in ref.[24]. In the present approach we choose the scheme that corresponds with the value of D which is identical with the value for the first order minimum cross diffusion scheme: D=min(a,b,c).

4.2.4 *3D Multidimensional Limiters*

In this section the 2D multidimensional limiters discussed before are extended for the 3D upwind schemes. The 2nd order zero cross diffusion scheme related to D=min(a,b,c) (fig.8) is rewritten as the first order minimum cross diffusion scheme plus anti-diffusive limited correction term,

$$d^{(2)}_{i+1/2,j,k} = d^{(1)}_{i+1/2,j,k}$$

$$+ \ (\Delta\alpha_x\, \delta u_{i,j-1/2,k} + \Delta\beta_x\, \delta u_{i,j,k-1/2})\, \Phi(r_{i+1/2,j,k}) \quad (24)$$

with $$\Delta\alpha_x = \alpha^{(2)}_x - \alpha^{(1)}_x = \frac{1}{2}\,(\,b - \min(a,b)\,)$$

$$\Delta\beta_x = \beta^{(2)}_x - \beta^{(1)}_x = \frac{1}{2}\,(\,c - \min(a,c)\,)$$

Remark that only one limiter is applied in (24). An alternative possibility would be to add a different limiter/ratio to each component of the correction term. Near discontinuities the limiter is switched off (Φ=0) and the 1st order multi-D dissipation is applied. In smooth regions Φ =1 and then the linear 2nd order scheme is applied.

Notice that this definition of L is not the same as for the linear multi-D models (20) because the reference dissipation has been changed to the minimum cross diffusion scheme instead of the classical 1st order upwind scheme. The definition of the 3D ratio is based on the variations δu_y and δu_z in the correction term of (24) and some extra variation in the third direction. Thus for face i+1/2,j,k a variation δu_x is introduced in the 3D ratio,

$$r_{i+1/2,j,k} = \frac{c_1\,\delta u_x + c_2\,\delta u_y + c_3\,\delta u_z}{\Delta\alpha_x \delta u_y + \Delta\beta_x \delta u_z} \qquad (25)$$

Equation (25) has the same form as the definiton of a 3D ratio in the formulation of a new fluctuation splitting scheme in ref.[26] . Different possibilities can be considered for δu_x, as shown in ref.[12] where three different classes of 3D ratios are defined. Each definition corresponds with the variations in a tetrahedron constructed by the three variations along x-,y- and z-axis. For more details concerning monotonicity conditions see ref.[12].

5. EXTENSION FOR THE EULER/NS EQUATIONS

The conservative form of the 3D Navier-Stokes equations is written as:

$$\frac{\partial U}{\partial t} + \frac{\partial}{\partial x}(F-F_v) + \frac{\partial}{\partial y}(G-G_v) + \frac{\partial}{\partial z}(H-H_v) = 0 \qquad (26)$$

with conservative variables U =(ρ,ρu,ρv,ρw,ρE)^t, the inviscid fluxes (F,G,H) and the viscous fluxes (F_v,G_v,H_v). The latter are appromimated by a central discretization and will not be considered below. Application of the multidimensional upwind dissipation models from section 4 to the inviscid fluxes consists of 3 consecutive steps:

1) *Characteristic decomposition of the Euler system*
2) *Multi-D discretisation of the characteristic equations*
3) *Re-transformation to conservative numerical flux.*

5.1. Characteristic Decomposition

5.1.1 *Characteristic variables/ compatibility equations*

The 2D Euler equations are expressed by

$$\frac{\partial U}{\partial t} + \vec{\nabla}.\vec{F} = \frac{\partial U}{\partial t} + \vec{A}.\vec{\nabla}U = 0 \qquad (27)$$

where $\vec{A}$=(A,B) are the jacobian matrices. The eigenvalues of the matrix $K = \vec{A}.\vec{\kappa}$ associated to an arbitrary unit propagation direction $\vec{\kappa}$ define for a large part the behaviour of the solutions to the Euler equations. Wave-like solutions exist if the eigenvalues of K are real and the corresponding eigenvectors linear independent[5]. The latter define a similarity transformation which diagonalizes matrix K,

$$P^{-1}\,(\,\vec{A}.\vec{\kappa}\,)\,P = \Lambda \qquad (28)$$

with the left eigenvectors being the rows of P^{-1}, the right eigenvectors being the columns of P and the diagonal matrix Λ consisting of the eigenvalues,

$$\lambda^{(1)}=\lambda^{(2)}= \vec{v}.\vec{\kappa}\ ,\ \lambda^{(3)}= \vec{v}.\vec{\kappa}+ c\ ,\ \lambda^{(4)}= \vec{v}.\vec{\kappa}- c \qquad (29)$$

Using the left eigenvectors, a set of characteristic variables can be constructed,

$$\delta W = P^{-1}\,\delta U \quad \text{or}\ \delta U = P\,\delta W = \sum_{k=1}^{4} \delta w^{(k)}\,\vec{r}^{\,(k)} \qquad (30)$$

or
$$\delta w^{(1)} = \delta\rho - \delta p/c^2$$
$$\delta w^{(2)} = \vec{l}_1.\delta\vec{v} + \mu(\delta\rho - \delta p/c^2)$$
$$\delta w^{(3)} = \vec{\kappa}_2.\delta\vec{v} + \delta p/\rho c \qquad (31)$$
$$\delta w^{(4)} = -\vec{\kappa}_3.\delta\vec{v} + \delta p/\rho c$$

with μ being a free parameter. Eq. (31) is not the only possible definition of characteristic variables[5], but the above choice is well appropriate for our purpose and is based on 3 arbitrary propagation directions,

$$\vec{\kappa}_i = (\kappa_{ix},\kappa_{iy}) = (\cos\theta_i,\sin\theta_i)\ ,\ \vec{l}_i = (\kappa_{iy},-\kappa_{ix})\ \ \text{for i=1,2,3} \quad (32)$$

In order to identify appropriate wave decompositions, the characteristic variables are defined by different propagation directions: $w^{(1)}$,$w^{(2)}$ are related to $\vec{\kappa}_1$ and $w^{(3)}$ and $w^{(4)}$ are related to respectively $\vec{\kappa}_2$,$\vec{\kappa}_3$. Multiplying eq. (27) by the matrix P^{-1} and introducing the characteristic variables (30)-(31) leads to the characteristic compatibility equations :

$$\frac{\partial W}{\partial t} + P^{-1}A\,P\,\frac{\partial W}{\partial x} + P^{-1}B\,P\,\frac{\partial W}{\partial y} = 0 \qquad (33)$$

or after working out (33) explicitly,

$$\frac{\partial w^{(1)}}{\partial t} + \vec{v}.\vec{\nabla}w^{(1)} = 0$$
$$\frac{\partial w^{(2)}}{\partial t} + \vec{v}.\vec{\nabla}w^{(2)} + \frac{1}{\rho}\,\vec{l}_1.\vec{\nabla p} = 0$$
$$\frac{\partial w^{(3)}}{\partial t} + (\vec{v} + c\vec{\kappa}_2).\vec{\nabla}w^{(3)} + c\,\vec{l}_2.(\vec{l}_2.\vec{\nabla})\vec{v} = 0 \qquad (34)$$
$$\frac{\partial w^{(4)}}{\partial t} + (\vec{v} - c\vec{\kappa}_3).\vec{\nabla}w^{(4)} + c\,\vec{l}_3.(\vec{l}_3.\vec{\nabla})\vec{v} = 0$$

The corresponding 4-wave model consists of one entropy wave, one shear wave and two acoustic waves[8]. The first two terms in each equation of (34) represent the convection of the associated wave in the characteristic direction,

$$\vec{a}_c^{(1)} = \vec{a}_c^{(2)} = \vec{v}\,, \quad \vec{a}_c^{(3)} = \vec{v} + c\vec{\kappa}_2\,, \quad \vec{a}_c^{(4)} = \vec{v} - c\vec{\kappa}_3 \qquad (35)$$

The subscript c in (35) refers to the convective part. The third terms in (34) are the coupling or source terms, and their presence results from the fact that the jacobian matrices in (27) are not simultaneously diagonalizable by matrix P. Notice that the coupling terms show also an advective behaviour associated to the directions,

$$\vec{a}_s^{(2)} = \vec{l}_1\,, \vec{a}_s^{(3)} = \vec{l}_2\,, \quad \vec{a}_s^{(4)} = \vec{l}_3 \qquad (36)$$

that are the the normal directions to the propagation directions $\vec{\kappa}_1, \vec{\kappa}_2, \vec{\kappa}_3$. The subscript s refers to the source terms.

5.1.2 *Propagation directions*

The choice of the propagation directions $\vec{\kappa}_1, \vec{\kappa}_2, \vec{\kappa}_3$ with related normals $\vec{l}_1, \vec{l}_2, \vec{l}_3$ in (34) are still to be defined. A main constraint on these directions comes from the factor $\vec{\kappa}_1.(\vec{\kappa}_2+\vec{\kappa}_3)$ that shows up in the denominator of P. To prevent an ill-defined transformation this factor should be maximized. Other conditions to impose on the design of the propagation directions are the continuity from subsonic to supersonic flow range and robustness. Different possibilities of propagation directions have been examined.

- *Diagonalization approach*

The source terms in the system of compatibility equations (34) can be eliminated by the following set of propagation directions[8],

$$\vec{l}_1.\vec{\nabla}p = 0\,, \quad \vec{l}_2.(\vec{l}_2.\vec{\nabla})\vec{v} = 0\,, \quad \vec{l}_3 = \vec{l}_2 \qquad (37)$$

The first direction $\vec{\kappa}_1$ is taken along the pressure gradient while $\vec{\kappa}_2, \vec{\kappa}_3$ are taken equal and defined by the strain rate tensor. The use of this set of directions depending on gradients of the solution, shows a lack of robustness in Euler calculations[7].

- *Combination pressure gradient/velocity*

In non-uniform regions $\vec{\kappa}_1, \vec{\kappa}_2, \vec{\kappa}_3$ are taken along the pressure gradient,

$$\vec{l}_1.\vec{\nabla}p = 0 \quad , \qquad \vec{l}_3 = \vec{l}_2 = \vec{l}_1 \qquad (38)$$

In smooth regions a continuous switch between the pressure gradient (38) and the streamline direction is introduced. This model shows good accuracy in both subsonic and supersonic regime[7,9,10]. A better convergence behaviour than with (37) is obtained especially in supersonic flow. In some cases (e.g. subsonic flows) convergence can only be obtained by freezing the directions after a certain residual drop of 1 or 2 orders[9,10]

- *Streamline direction*

A much simpler choice is taking the directions along the streamline

$$\vec{l}_1.\vec{v} = 0 \quad , \qquad \vec{l}_3 = \vec{l}_2 = \vec{l}_1 \qquad (39)$$

This choice seems to have good convergence behaviour but has a poor accuracy especially in supersonic inviscid flows near discontinuities[9]. On the contrary very good results are obtained in a laminar boundary layer on very coarse meshes[10].

- *Convection of entropy and enthalpy*

The first characteristic direction $\vec{\kappa}_1$ is taken perpendicular to the velocity

$$\vec{l}_1 = \frac{\vec{v}}{\|\vec{v}\|} \qquad (40)$$

Using the definition of specific entropy and total enthalpy,

$$\delta S = \frac{\gamma c_v}{\rho}\left(\frac{\delta p}{c^2} - \delta\rho\right), \quad \delta H = \frac{\gamma c^2}{\rho(\gamma\text{-}1)}\left(\frac{\delta p}{c^2} - \frac{\delta\rho}{\gamma}\right) + \vec{v}.\delta\vec{v} \qquad (41)$$

the first two characteristic equations of (34) are rewritten as

$$\frac{\partial w^{(1)}}{\partial t} - \frac{\rho}{\gamma c_v}\vec{\mathbf{v}}.\vec{\nabla}\mathbf{S} = 0$$
$$\frac{\partial w^{(2)}}{\partial t} + \frac{\vec{v}}{\|\vec{v}\|}.\left(\vec{\nabla}\mathbf{H} - \frac{\rho}{\gamma c_v}\left(\mu\|\vec{v}\| + \frac{c^2}{\rho(\gamma-1)}\right)\vec{\nabla}\mathbf{S}\right) = 0 \qquad (42)$$

Eq. (42) shows that in steady state the entropy and total enthalpy are constant along the streamline, see also ref.[17,18,19] (where μ=0). The 2nd equation of (42) can further be simplified by choosing the parameter μ as $\mu = -c^2/\rho\|\vec{v}\|(\gamma\text{-}1)$ leading to

$$\frac{\partial w^{(1)}}{\partial t} - \frac{\rho}{\gamma c_v}\vec{\mathbf{v}}.\vec{\nabla}\mathbf{S} = 0$$
$$\frac{\partial w^{(2)}}{\partial t} + \frac{1}{\|\vec{v}\|}\vec{\mathbf{v}}.\vec{\nabla}\mathbf{H} = 0 \qquad (43)$$

As a result the Euler system (34) is splitted in a hyperbolic part that represents the convection of entropy and enthalpy along the streamline in steady state (43) and a remaining acoustic subsystem with source terms, as in the hyperbolic/elliptic splitting in ref.[16,18].

- *Machangle splitting*

In the framework of the fluctuation splitting schemes[17,18,19] a machangle splitting was developed. The first direction is taken perpendicular to the velocity and the 2nd and 3rd directions are respectively perpendicular to the positive and negative characteristics or machlines

$$\theta_1 = \theta + \frac{\pi}{2}, \quad \theta_2 = \theta_1 + \mu\,, \quad \theta_3 = \theta_1 - \mu \qquad (44)$$

with θ the flow angle and $\mu = \arctan\left(1/\sqrt{M^2-1}\right)$ the machangle. A fully decouple system of characteristic equations is obtained in steady state,

$$\vec{v}.\vec{\nabla}S = 0 \qquad \vec{v}.\vec{\nabla}H = 0$$
$$(\vec{v} + c\vec{\kappa}_2).\vec{\nabla}R^3 = 0 \quad , \quad (\vec{v} - c\vec{\kappa}_3).\vec{\nabla}R^4 = 0 \qquad (45)$$

using the steady Riemann variables:

$$\delta R^3 = \delta w^{(3)} + \frac{c}{\|\vec{v}+c\vec{\kappa}_2\|}(\vec{l}_2.\delta\vec{v}) \qquad (46a)$$

$$\delta R^4 = \delta w^{(4)} + \frac{c}{\|\vec{v}-c\vec{\kappa}_3\|}(\vec{l}_3.\delta\vec{v}) \qquad (46b)$$

- *Pseudo-Machangle splitting*

This model is an algebraic continuation of the supersonic machangle decomposition in the subsonic range with continuity at M=1, developed in the fluctuation splitting approach[17,18,19]. The corresponding directions are,

$$\theta_1 = \theta_2, \quad \theta_2 = \theta \pm \mu \pm \frac{\pi}{2}, \quad \theta_3 = \theta_2 \qquad (47)$$

with $\mu = \arctan\left(1/\sqrt{|M^2 - 1|}\right)$ defined as the pseudo-machangle. Notice the 2 sets of propagation directions leading to 2 splittings of the residual. For the final residual the average of both splittings is taken.

Considerable research is still being performed to identify the most suitable directions, see for instance[17,19] for a recent survey.

5.2. Numerical Flux Formulation

The space operators in the characteristic compatibility equations (34),(43) and (45) discussed in section 5.1 are discretised. In the case that there is no source term, the space operator is expressed by

$$\text{no source term :} \quad \vec{a}_c^{(k)} \,.\, \vec{\nabla} w^{(k)} \quad \text{or} \quad \vec{a}_c^{(k)} \,.\, \vec{\nabla} R^{(k)} \qquad (48)$$

where the gradient acts on the characteristic variable or a steady Riemann invariant. When a source term is present, the space operator can be written as,

$$\text{with source term :} \quad \vec{a}_c^{(k)} \,.\, \vec{\nabla} w^{(k)} + \vec{a}_s^{(k)} \,.\, \vec{\nabla} s^{(k)} \qquad (49)$$

where the convective and source terms are written as an advection of respectively a characteristic variable and a 'source' variable along the associated directions (35) and (36). In both cases the source and convective terms have the same form and can be treated by the same multi-D scheme or dissipation model.

In the formulation used in previous work[10,12,31,33] the scalar multi-D dissipation models were applied only to the convective terms while the source terms were discretised by a central approximation without artificial dissipation. In the present approach also the source terms can be treated with a multi-D scheme based on the associated speed (36).

Discretising both convective and source terms leads to two numerical fluxes or dissipations for every scalar equation. Next the scalar multi-D discretisation is re-transformed to the conservative residual by use of the right eigenvectors. The resulting inviscid numerical flux on cell face i+1/2,j is defined by

$$F^*_{i+1/2,j} = \frac{1}{2}\left(F_{i,j} + F_{i+1,j} \right) - \sum_{k=1}^{4} \left(d_c^{(k)} + d_s^{(k)} \right) \vec{r}^{\,(k)} \qquad (50)$$

where d_c and d_s represent respectively the scalar multi-D dissipation of the convective and source part for each of the 4 characteristic equations. The old formulation where the convective term is treated by a multi-D scheme and the source term by a central scheme without artificial dissipation is easily recovered by putting the dissipation for the source term in (50) to zero.

6. RESULTS

6.1. 2D supersonic Laval nozzle

The inviscid supersonic flow in a Laval nozzle is calculated at a Machnumber of 2.91 on an H-type mesh with 128x32 cells. The first order minimum cross diffusion scheme (4MCD) and 5-point continuous zero cross diffusion scheme (5ZCD) combined with minmod limiter and the ratio of subclass (Ia) is investigated. The multi-D schemes are compared with the classical 2nd order Flux Difference Splitting scheme (FDS2) with minmod limiter. The classical scheme is tested on a finer mesh of 256x64 cells, as reference solution. The extension to the Euler equations is based on the 2D characteristic decomposition (34) with the 3 characteristic directions $\vec{\kappa}_1, \vec{\kappa}_2, \vec{\kappa}_3$ defined by the machangle splitting. Both the convective and source terms of (34) are discretised with the same scalar multi-D dissipation model.

Figure 9 shows the isomachlines for the 4 solutions. The first order multi-D scheme performs well in comparison with the classical 2nd order scheme up to the 2nd reflection of the shock structure. The 2nd order multi-D scheme is superior to the classical scheme on the same mesh. It compares very well with the reference solution on the finer mesh. Figure 10 and 11 show respectively the Machnumber and total temperature distribution along the symmetry-axis. The total temperature or total enthalpy should be constant in the whole field. The errors for the multi-D schemes are much smaller than for the classical results, even on the finer mesh.

Figure 12 shows the convergence history. Both first and second order 2D results show a good convergence behaviour obtained with a 3 level multigrid acceleration combined with a 5-stage Runge Kutta prodecure and residual smoothing with a CFL of respectively 10.0 and 8.0.

6.2. 3D supersonic corner flow

An inviscid supersonic corner flow[4] (M=3.0) is considered, which is generated by two unswept compression ramps with 9.5 deg. wedge angle as illustrated in figure 13. The first order 3D minimum cross diffusion scheme is tested in comparison with classical first and second order (minmod limiter) upwind schemes on a uniform mesh with 32x32x32 cells. The accuracy of the 3D scheme is investigated for both 3D and 2D flow phenomena appearing in this testcase. The extension to the Euler equations is performed using the 3D extension of the characteristic variables (31) and equations (34), see ref.[12]. The three characteristic directions $\vec{\kappa}_1, \vec{\kappa}_2, \vec{\kappa}_3$ are taken along the pressure gradient direction. When the pressure gradient goes to zero a blending is performed with the velocity direction.

Figure 14 shows the convergence history. No freezing of the directions was needed to reach convergence with the 3D scheme. Convergence is obtained with the use of multigrid acceleration and residual smoothing with a 5 stage Runge Kutta procedure with CFL =10.

Isomach lines are shown in figure 15. The classical first order scheme shows smeared out shocks and no contact discontinuities while the first order 3D result shows an accuracy comparable with classical 2nd order. From the isomachlines near the solid walls one can conclude that the multi-d result shows less entropy creation than the classical schemes.

7. CONCLUSIONS

Genuinely 3D/2D multidimensional upwind schemes are developed for the Euler/Navier-Stokes equations. The schemes are formulated in the framework of dimensional-split central or upwind dissipation models, leading to a new concept of compact 3D/2D multidimensional upwind dissipation.

A unification of 2D compact linear schemes is shown based on two classes of 6-point stencils. Each class has a two-parameter subclass of zero cross diffusion schemes and a unique second order scheme. Both families have a 4-point subclass in common that has a unique minimum cross diffusion scheme and a zero cross diffusion scheme.

A class of 3D scalar convection schemes based on an 8-point compact stencil is derived that reduces to the 4-point subclass in 2D. It has a unique first order scheme with minimum cross diffusion and a one-parameter subclass of zero cross diffusion schemes being second order accurate for a homogeneous convection equation.

Second order monotone schemes are explored. The dissipation is written as the 1st order minimum cross diffusion dissipation plus additional anti-diffusive terms. Three- and two-dimensional limiters, depending on ratios of flux differences in different mesh directions, are introduced. In 2D two families of ratios related to two types of triangles are defined. In each class two sub-families are considered related to variations along the mesh line or along a diagonal.

Extensions to the Euler-Navier/Stokes equations are obtained through a characteristic decomposition using characteristic variables with 3 different propagation directions. A review is given of different choices for the directions. For supersonic flow the combination of pressure gradient and velocity seems to be an accurate choice but the machangle splitting is more robust. Application of the 3D minimum cross diffusion scheme in combination with the pressure gradient approach to a 3D supersonic testcase shows comparable accuracy with a classical 2nd order dimensional-split upwind scheme.

For subsonic and supersonic flow the streamline direction is not yet accurate enough and so research is still needed to identify more effective choices.

A new formulation is introduced where both the convective and source terms can be discretised with a multi-D scheme using its associated characteristic speed. Application of this new approach in combination with the machangle splitting directions to a 2D supersonic testcase shows better accuracy with lower total temperature error than classical 2nd order dimensional-split upwind schemes.

ACKNOWLEDGEMENTS

This research is supported by the Commission of the European Community under Contract AERO-CT-0040/PL-2037 in Area 5 (Aeronautics) of the Brite/Euram Programme.

REFERENCES

[1] Dadone, A., and Grossman, B. (1991). 'A rotated upwind scheme for the Euler equations.' *AIAA Paper 91-0635.*

[2] Deconinck, H., Struijs, R., Bourgois, G., and Roe, P.L. (1994). 'High Resolution Shock Capturing Cell Vertex Advection Schemes on Unstructured Grids.' *VKI Lecture Series 1994-05 on Computational Fluid Dynamics*, Von Karman Institute, Brussels, Belgium.

[3] Deconinck, H., Struijs, R., and Roe, P. L. (1991). 'Fluctuation splitting for multidimensional convection problem: an alternative to finite volume and finite element methods.' *VKI Lecture Series 1990-3 on Computational Fluid Dynamics*, Von Karman Institute, Brussels, Belgium.

[4] Hentschel, R., and Hirschel, E.H. (1994). 'Self Adaptive Flow Computations on Structured Grids.' *Proc. ECCOMAS 2nd European Computational Fluid Dynamics Conference*, Wiley, September 1994,pp.242-249.

[5] Hirsch, Ch. (1990). 'Numerical Computation of Internal and External Flows.' Vol. 2, John Wiley & Sons, Chichester, England.

6 Hirsch, Ch. (1991). 'A General Analysis of Two-Dimensional Convection Schemes.' *VKI Lecture Series 1991-02 on Computational Fluid Dynamics*, Von Karman Institute, Brussels, Belgium.

[7] Hirsch, Ch., and Lacor, C. (1989). 'Upwind Algorithms Based on a Diagonalization Procedure for the Multidimensional Euler Equations.' *AIAA Paper 89-1958,* AIAA 9th Computational Fluid Dynamics Conference.

[8] Hirsch, Ch., Lacor, C., and Deconinck, H. (1987). 'Convection Algorithms Based on a Diagonalization Procedure for the Multidimensional Euler Equations.' Proc. *AIAA Paper 87-1163, pp.667-676,* AIAA 8th Computational Fluid Dynamics Conference.

[9] Hirsch, Ch., and Van Ransbeeck, P. (1992). 'Cell-centered Multidimensional Upwind Algorithms and Structured Meshes.' *Proc. ECCOMAS 1st European Computational Fluid Dynamics Conference*, Elsevier, Vol. 1, pp. 53-60.

10 Hirsch, Ch., and Van Ransbeeck, P. (1994). 'Multidimensional Upwinding and Artificial Dissipation.' Published in Frontiers of Computational Fluid Dynamics, *Proceedings for the symposium Computing the Future: Advances and Prospects for Computational Aerodynamics*, Wiley, pp. 597-626.

[11] Hirsch, Ch., and Van Ransbeeck, P. (1995). 'A General Analysis of Multidimensional Convection Schemes.' Appendix in 'State of the Art of CFD in Industry' (Ch. Hirsch), *VKI Lecture Series 1995-07 on Industrial Computational Fluid Dynamics*, Von Karman Institute, Brussels, Belgium, pp.61-88.

12 Hirsch, Ch., and P. Van Ransbeeck (1995). 'Multidimensional Upwinding and Artificial Dissipation for the Euler/Navier-Stokes Equations.' *AIAA Paper 95-1702, Proc. AIAA 12th Computational Fluid Dynamics Conference.*

[13] Jameson, A. (1984). 'A Non-Oscillatory Shock Capturing Scheme using Flux Limited Dissipation.' MAE Report 1653, Princeton University, New Jersey. Published in *Lectures in Applied Mathematics*, Enquist, Osher and Somerville Eds, A.M.S., Part I, pp 345-370, 1985.

[14] Jameson, A. (1993). 'Artificial Diffusion, Upwind Biasing, Limiters and their Effect on Accuracy and Multigrid Convergence in Transonic and Hypersonic Flows.' *AIAA Paper 93-3359, AIAA 11th Computational Fluid Dynamics Conference*, Orlando.

[15] Levy, D.W., Powell, K.G., and van Leer, B. (1989). 'An Implementation of a Grid-Independent Upwind Scheme for the Euler Equations.' *AIAA Paper 89-1931-CP, Proceedings*

AIAA 9th Computational Fluid Dynamics Conference, Buffalo,.pp. 8-24.

[16] Mesaros, L.M., and Roe P.L. (1995). 'Multidimensional Fluctuation Splitting Schemes Based on Decomposition Methods.' *AIAA Paper 95-1699, Proc. AIAA 12th Computational Fluid Dynamics Conference,* San Diego.

[17] Paillère, H, Carette, J.C., and Deconinck, H. (1994). 'Multidimensional Upwind and SUPG methods for the Solution of the Compressible Flow Equations on Unstructured Grids.' *VKI Lecture Series 1994-05 on Computational Fluid Dynamics*, Von Karman Institute, Brussels, Belgium.

[18] Paillère, H, Deconinck, H., and Roe P.L. (1995). 'Conservative upwind residual-distribution schemes based on the steady characteristics of the Euler equations.' *AIAA Paper 95-1700, Proc. AIAA 12th Computational Fluid Dynamics Conference,* San Diego.

[19] Paillère, H. (1995). 'Multidimensional Upwind Residual Distribution Schemes for the Euler and Navier-Stokes Equations on Unstructured Grids.' Ph.D. Thesis, Von Karman Institute, Brussels, Belgium.

[20] Ricc, J.G.,and Schnipke, R.J.(1985). 'A Monotone Streamline Upwind Finite Element Method for Convection Dominated Flows.' *Computer Methods in Appl. Mech. and Engineering*, Vol. 48, pp.313-327.

[21] Roe, P.L. (1986). 'Discrete Models for the numerical analysis of time-dependent multidimensional gas dynamics.' *Journal of Computational Physics*, Vol.63.

[22] Roe, P.L. (1990). ''Optimum' upwind advection on triangular mesh.' *ICASE Report No. 90-75.*

[23] Roe, P.L. (1994). 'Multidimensional Upwinding-Motivation and Concepts.' *VKI Lecture Series 1994-05 on Computational Fluid Dynmics*, Von Karman Institute, Brussels, Belgium.

[24] Roe, P.L., and Sidilkover, D. (1992). 'Optimum Positive Linear Schemes for Advection in Two and Three Dimensions.' *SIAM J. Numer. Anal.*,Vol. 29, No.6, pp. 1542-1568.

[25] Sidilkover, D. (1989). 'Numerical Solution to Steady-State Problems with Discontinuities.' *Ph.D Thesis* . Dept. Mathematics, Weizmann Institute of Science, Israel.

[26] Sidilkover, D. (1994). 'A genuinely multidimensional upwind scheme and efficient multigrid solver for the compressible Euler equations.' Submitted for Journal of Computational Physics.

[27] Sidilkover, D., and Roe, P.L. (1994). 'Unification of some advection schemes in two dimensions.' Submitted for publication.

[28] Spekreyse, S.P. (1987). 'Multigrid Solution of Monotone Second order Discretizations of Hyperbolic Conservation Laws.' *Math. Comp.*,vol.49,pp.135-155.

[29] Swanson, R.C., and Turkel, E. (1992). 'On Central-Difference and Upwind Schemes.' *Journal of Computational Physics*, 101, pp.297-306.

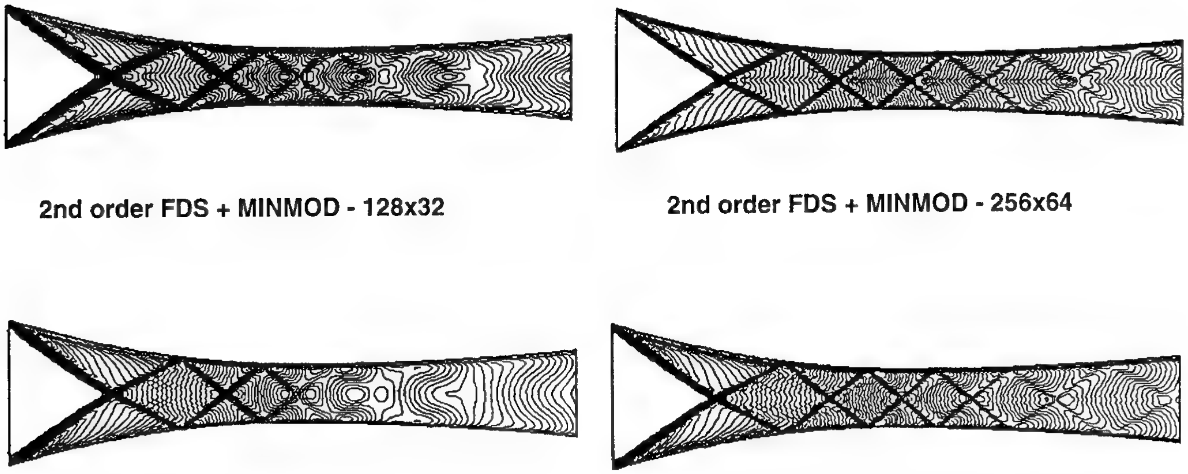
[30] Van Leer, B. (1992). 'Progress in multi-dimensional upwind differencing.' *ICASE Report No. 92-43.*

[31] Van Ransbeeck, P., and Hirsch, Ch. (1993). 'New Upwind Dissipation Models with a Multidimensional Approach.' *AIAA Paper 93-9304, Proc. AIAA 11th Computational Fluid Dynamics Conference*, pp. 81-91.

[32] Van Ransbeeck, P., and Hirsch, Ch. (1994). 'Solution Adaptive Navier-Stokes Solvers using multidimensional upwind schemes and multigrid acceleration.' VUB part of Progress Report Brite/Euram contract AERO-CT-0040/PL-2037, VKI CR 1994-26.

[33] Van Ransbeeck, P., and Hirsch, Ch. (1994). 'Multidimensional Upwind Dissipation Models for the 2D Navier-Stokes Equations.' *Proc. ECCOMAS 2nd European Computational Fluid Dynamics Conference*, Wiley, September 1994,pp. 655-662.

FIGURES



2nd order FDS + MINMOD - 128x32

2nd order FDS + MINMOD - 256x64

Multi-D 1st order - 128x32

Multi-D 2nd order + MINMOD + RATIO IA - 128x32

Figure 9 Supersonic Laval nozzle (M=2.91), isomachlines,1.41<M<2.91, ΔM=0.02.

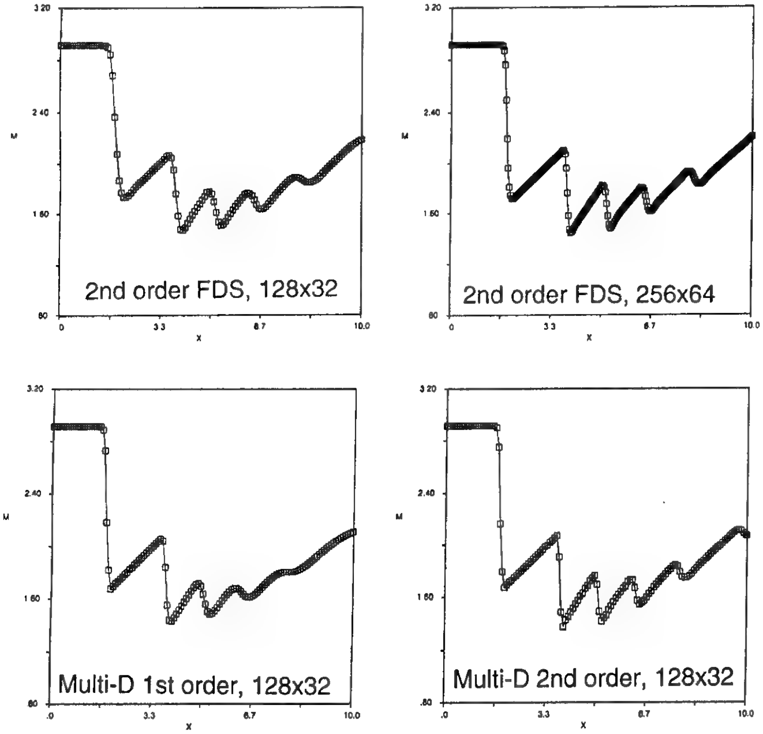


Figure 10 Supersonic Laval nozzle (M=2.91), Machnumber distribution along symmetry-axis.

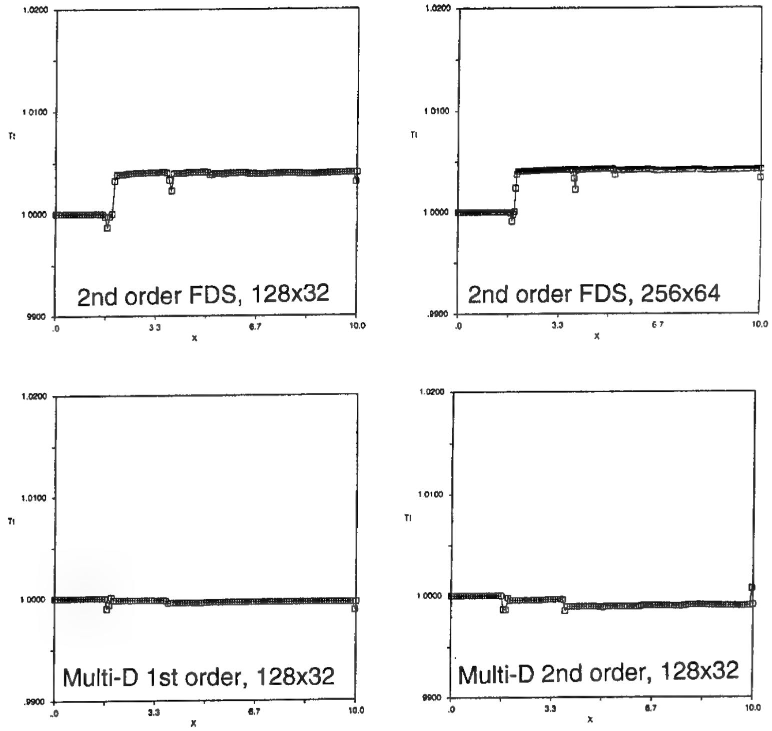


Figure 11 Supersonic Laval nozzle (M=2.91), Total temperature distribution along symmetry-axis.

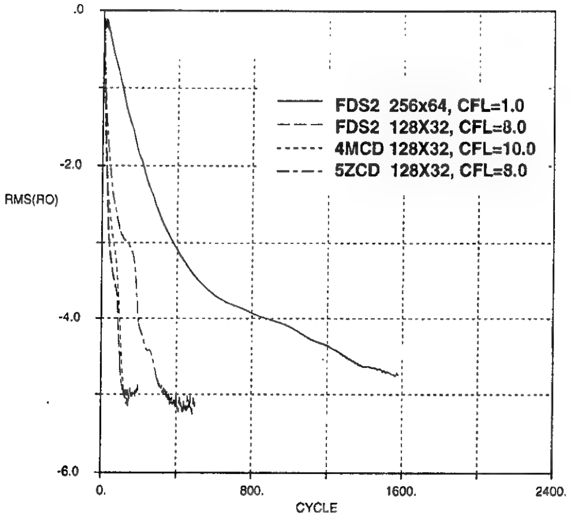
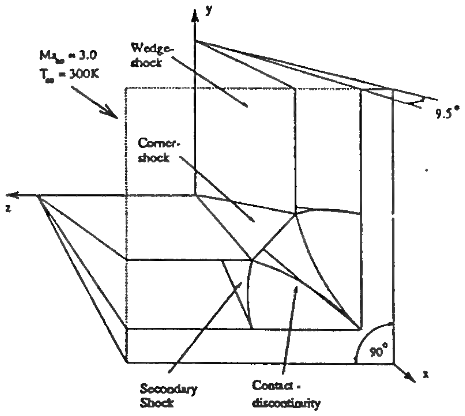


Figure 12 Supersonic Laval nozzle (M=2.91), Convergence history.



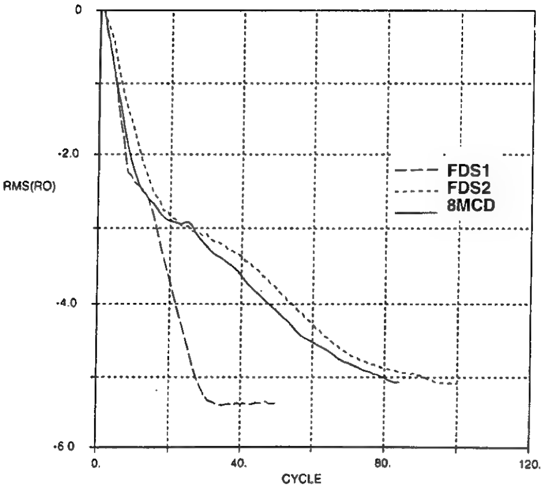
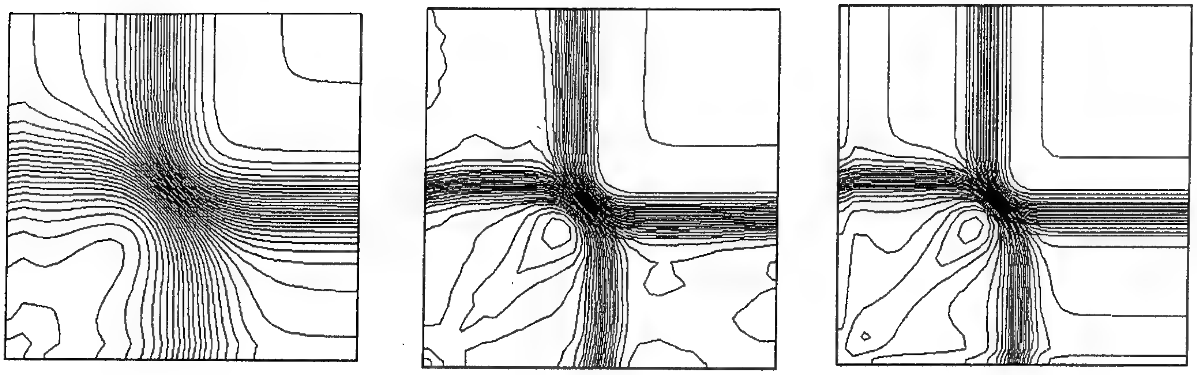


Figure 13 Supersonic corner flow, flow behaviour[4]. **Figure 14** Supersonic corner flow, convergence history RMS.



(a) FDS1

(b) 8MCD

(c) FDS2-Minmod limiter

Figure 15 Supersonic corner flow, isomachline comparison (2.0< M< 3.0, ΔM=0.02).

A PCG/E-B-E ITERATION FOR HIGH ORDER AND FAST SOLUTION OF 3-D NAVIER-STOKES EQUATIONS

A. Rüstem Aslan, Ülgen Gülçat and Aydın Mısırlıoğlu
Faculty of Aeronautics and Astronautics
ITU, 80626, Maslak, Istanbul, Turkey

SUMMARY

A second order accurate (both in time and space) an explicit/implicit scheme is implemented for the solution of three-dimensional incompressible Navier-Stokes equations involving high Reynolds Number flows about complex configurations. A fourth order accurate artificial dissipation term on the momentum equations are used for stabilizing. Finite Element Method (FEM) with an explicit time marching scheme is used for the solution, and element by element (E-B-E) technique is employed in order to ease the memory requirements needed by the storage of the stiffness matrix of FEM. The cubic cavity problem, laminar flow past a sphere at a high Reynolds number and an incompressible viscous flow around the fuselage of a helicopter are succesfully solved using the first and the second order accurate schemes. Comparison of the results are also provided.

1. INTRODUCTION

Recent advances in iterative solution techniques enabled CFD researchers to solve large scale problems in acceptable computation times with the fast processors of 90's. The iterative solvers have become the CFD's convenient tools which do not require excessive memories on computers for either implicit time marching schemes or inversion of elliptic equations. For finite element computations, element by element (E-B-E) iteration schemes demand the least amount of memory. The conjugate gradient (CG) method, which is the Krylov subspace technique applied on symmetric operators, becomes an efficient, indeed the fastest converging, iterative method when applied with preconditioning (PCG) to the discrete form of the equations.

During the last two decades, solution of three-dimensional Navier-Stokes equations received considerable attention. However, for a numerical technique to fulfill the demands of 90's, the accuracy of the scheme must be at least second order for both in time and space discretizations.

In this study, a second order accurate (both in time and space) an explicit/implicit scheme is implemented for the solution of three-dimensional incompressible Navier-Stokes equations involving high Reynolds Number flows about complex configurations. To do so, a fourth order accurate artificial dissipation term on the momentum equations are used for stabilizing. Finite Element Method (FEM) with an explicit time marching scheme is used for the solution, and element by element (E-B-E) technique is employed in order to ease the memory requirements needed by the storage of the stiffness matrix of FEM[1]. Since the scheme is time accurate, the transient nature of the flow field is properly resolved.

For the calibration of the code the cubic cavity problem is solved using the first and the second order accurate schemes. The comparison with the existing literature[2] is satisfactory even for a coarse grid. The solution with the fourth order artificial viscosity adequately resolves the recirculating region as opposed to the solution with the second order artificial viscosity.

As the second study, laminar flow past a sphere at a high Reynolds number, Re = 162 000, is solved with the both schemes. Finally, in order to test the capabilities of the code, an incompressible viscous flow of Re=50 000, around the fuselage of a helicopter is studied.

All the computations are performed on a personal computer equipped with a i860 Number Smasher board with 32 Mbytes of memory.

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

2. FORMULATION

2.1 Governing Equations

The Navier-Stokes and the continuity equations for the unsteady, incompressible flow of a viscous fluid, in the absence of body forces are:

$$\frac{D\mathbf{V}}{Dt} = -\nabla p + \frac{1}{Re}\nabla^2\mathbf{V} \qquad (1)$$

$$\nabla \cdot \mathbf{V} = 0 \qquad (2)$$

The equations are written in vector form (boldface type symbols denote vector or matrix quantities). The variables are non-dimensionalized using a reference velocity and a characteristic length, as usual. Re is the Reynolds number, $Re = Ul/\nu$ where U is the reference velocity, l is the characteristic length and ν is the kinematic viscosity of the fluid. $\mathbf{V}$ corresponds to the Cartesian velocity components u,v, and w. Pressure is symbolized with p and the time is with t.

For a well posed problem, the governing equations are complemented with the following initial (t=0)

$$\mathbf{V}(\mathbf{x},0) = \mathbf{V^o}(\mathbf{x}) \quad \text{and} \quad p(\mathbf{x},0) = p^o(\mathbf{x}) \qquad (3)$$

and boundary conditions which have to be specified on related surfaces:

$$\mathbf{V} = \mathbf{G} \quad \text{and} \quad -p\mathbf{n} + \frac{1}{Re}(\nabla\mathbf{V})\cdot\mathbf{n} = \mathbf{F} \qquad (4)$$

where $\mathbf{x}$ is the position vector, $\mathbf{G}$ and $\mathbf{F}$ are prescribed boundary values, and $\mathbf{n}$ is the unit vector normal to the boundary.

2.2 Numerical Methods

The governing equations are integrated in time using both first and second order accurate schemes. The first order scheme follows that of [3] which constitutes a time marching scheme based on Helmholtz decomposition. A potential function with a single degree of freedom at each node is introduced and a Poisson equation for the potential is directly discretized. Eigth-node isoparametric brick elements and trilinear interpolation functions for the velocity and the auxiliary potential are used. The pressure is defined at the centroid of each element. In contrast to the potential and velocity, pressure values are interpolated using piecewise constant functions at each element. Application of the conventional Galerkin integral[4] to the equations and the boundary conditions gives integral finite element formulations for one brick element[3,5].

2.2.1 First order explicit formulation

Let $\mathbf{V_1}$ and $\mathbf{V_2}$ denote following velocity differences in vector form:
$\mathbf{V}_1 = \mathbf{V}^{m+\frac{1}{2}} - \mathbf{V}^m$
$\mathbf{V}_2 = \mathbf{V}^{m+1} - \mathbf{V}^{m+\frac{1}{2}}$
Using a forward difference operator for the time derivative in equation (1) and letting $\mathbf{V}^m$ and p^m be solutions at the known time level m, the first order explicit fractional step algorithm, over a single time step and in fully discrete matrix form, is given by in α direction as follows:

$$\frac{\mathbf{M}}{\Delta \mathrm{t}}\mathbf{V}_1^\alpha = \left[\mathbf{B}_\alpha + \mathrm{P_e}\mathbf{C}_\alpha - \left(\frac{\mathbf{A}}{\mathrm{Re}} + \mathbf{D}\right)\mathbf{V}_\alpha\right]^{\mathrm{m}} \qquad (5)$$

$$\mathbf{A}\ \phi = \mathbf{E}_\alpha\ \mathbf{V}_\alpha^{\mathrm{m}+\frac{1}{2}} \qquad (6)$$

$$\mathbf{M}\ \mathbf{V}_2^\alpha = \mathbf{E}_\alpha\ \phi \qquad (7)$$

$$p_e^{m+1} = p_e^m - \frac{\phi_e}{\Delta t} \qquad (8)$$

where ϕ is the auxiliary potential function, $\mathbf{M}$ is the lumped element mass matrix, $\mathbf{D}$ is the advection matrix, $\mathbf{A}$ is the stiffness matrix, $\mathbf{C}$ is the coefficient matrix for pressure, $\mathbf{B}$ is the vector due to boundary conditions and $\mathbf{E}$ is the matrix which arises due to incompressibility. Element potential ϕ_e is defined as

$$\phi_e = \frac{1}{vol(\Omega_e)}\int_{\Omega_e} N_i\ \phi_i\ d\Omega_e, \qquad i = 1,8 \qquad (9)$$

where Ω is the flow region to be solved, Γ is the boundary of Ω and N_i are the shape functions. Details of the formulation can be found in [5].

2.2.2 Second order explicit formulation

The second order time accurate scheme is somewhat similar to that of [6] wherein a new intermediate velocity field is introduced. Both explicit and implicit versions of the algorithm are devised. The explicit formulation resembles the first order explicit scheme except that the fractional step velocities are calculated in two steps. Let $\mathbf{V}_1$, $\mathbf{V}_2$ and $\mathbf{V}_3$ denote following velocity

vector differences:
$\mathbf{V}_1 = \mathbf{V}^{m+\frac{1}{2}} - \mathbf{V}^m$
$\mathbf{V}_2 = \mathbf{V}^* - \mathbf{V}^m$
$\mathbf{V}_3 = \mathbf{V}^{m+1} - \mathbf{V}^*$
the second order explicit fractional step algorithm, in fully discretised matrix form, over a single time step is defined by:

$$\frac{2\mathrm{M}}{\Delta \mathrm{t}}\mathbf{V}_1^\alpha = \left[\mathbf{B}_\alpha + \mathrm{P_e}\mathbf{C}_\alpha - \left(\frac{\mathbf{A}}{\mathrm{Re}} + \mathbf{D}\right)\mathbf{V}_\alpha\right]^{\mathrm{m}} \quad (10)$$

$$\frac{\mathrm{M}}{\Delta \mathrm{t}}\mathbf{V}_2^\alpha = \mathrm{P_e^m}\mathbf{C}_\alpha + \left[\mathbf{B}_\alpha - \left(\frac{\mathbf{A}}{\mathrm{Re}} + \mathbf{D}\right)\mathbf{V}_\alpha\right]^{\mathrm{m}+\frac{1}{2}} \quad (11)$$

$$\tfrac{1}{2}\mathbf{A}\phi = \mathbf{E}_\alpha \mathbf{V}_\alpha^* \quad (12)$$

$$\mathbf{M}\mathbf{V}_3^\alpha = \tfrac{1}{2}\mathbf{E}_\alpha \phi \quad (13)$$

$$p_e^{m+1} = p_e^m - \frac{\phi_e}{\Delta t} \quad (8)$$

The factor $\frac{1}{2}$ appearing in (12) and (13) is used for second order accuracy in time. In the formulation given above, $\mathbf{V}^*$ is a velocity vector which is not selonoidal.

2.2.3 Second order implicit formulation

The implicit fractional step formulation follows the same steps as does the explicit one. However, the formulation is obtained by adopting a Crank-Nicolson representation for the diffusion terms, but otherwise retaining the explicit formulation as before.
Using the same velocity diffrence formulas defined for the second order explicit formulation above, the second order implicit Galerkin fractional step algorithm, in fully discretised matrix form, over a single time step is defined by:

$$\left(\frac{2\mathrm{M}}{\Delta \mathrm{t}} + \frac{\mathbf{A}}{2\mathrm{Re}}\right)\mathbf{V}_1^\alpha = \quad (14)$$

$$\left[\mathbf{B}_\alpha + \mathrm{P_e}\mathbf{C}_\alpha - (\frac{\mathbf{A}}{\mathrm{Re}} + \mathbf{D})\mathbf{V}_\alpha\right]^{\mathrm{m}}$$

$$\left(\frac{\mathrm{M}}{\Delta \mathrm{t}} + \frac{\mathbf{A}}{2\mathrm{Re}}\right)\mathbf{V}_2^\alpha = \quad (15)$$

$$\left(\mathbf{B}_\alpha + \mathrm{P_e}\mathbf{C}_\alpha - \frac{\mathbf{A}}{\mathrm{Re}}\mathbf{V}_\alpha\right)^{\mathrm{m}} - (\mathbf{D}\ \mathbf{V}_\alpha)^{\mathrm{m}+\frac{1}{2}}$$

$$\tfrac{1}{2}\mathbf{A}\phi = \mathbf{E}_\alpha\ \mathbf{V}_\alpha^* \quad (16)$$

$$\left(\frac{\mathrm{M}}{\Delta \mathrm{t}} + \frac{\mathbf{A}}{2\mathrm{Re}}\right)\mathbf{V}_3^\alpha = \frac{1}{2\Delta \mathrm{t}}\mathbf{E}_\alpha\ \phi \quad (17)$$

$$p_e^{m+1} = p_e^m - \frac{\phi_e}{\Delta t} \quad (8)$$

For the implicit solutions of equations (14)-(17), Element By Element (E-B-E) technique[1] is employed in order to ease the memory requirements needed by the storage of the stiffness matrix of FEM. The iterative solution is fully vectorized[7]. The right hand side values of these equations are scaled with the square of the time step to increase accuracy. These scaling is found to reduce the number of iterations by almost 50%.

2.3 Artificial Dissipation

In the present study, a fourth order accurate artificial dissipation term on the momentum equations are used for stabilizing. The diffusion term is added explicitly to the right hand side of equation (1). Formulation given in reference[8] is extended to three-dimensions. The artificial viscosity term is computed in two steps at element level. First a second-order differencing is accomplished:

$$D_i^2 = \sum_{j=1}^{8} V_j - 8V_i$$

These values give the second-order distributions to cell corners (i) for the momentum equations. Then, fourth order distributions to cell corners are formed using the above values:

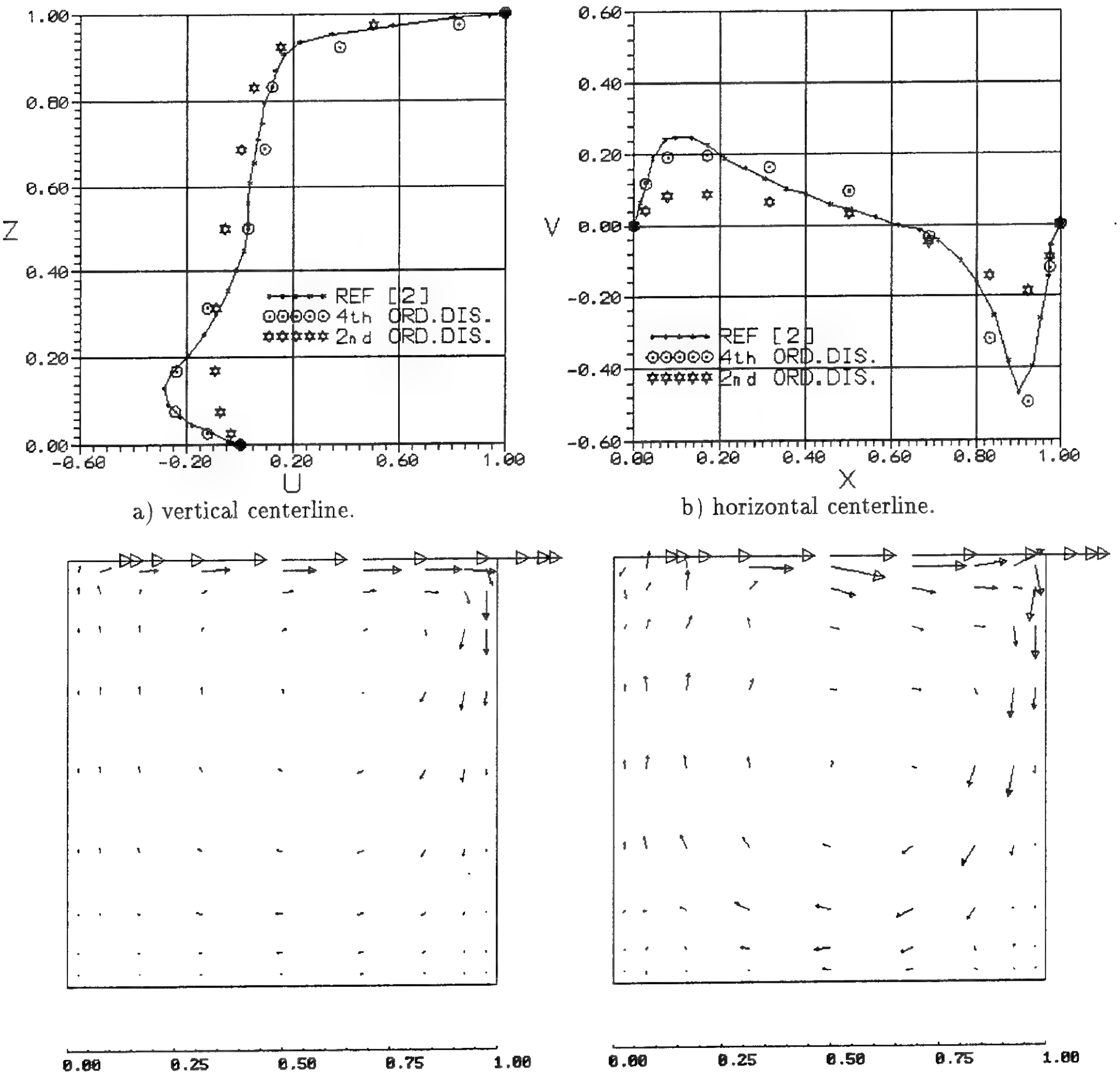
$$D_i^4 = \sum_{j=1}^{8} D_j^2 - 8D_i^2$$

These fourth order viscosity terms are multiplied by a certain coefficient when added to the momentum equations. All the velocity components are multiplied by the same coefficient, $c \leq 1/24$. No dissipation term is added to the poisson equation for the potential.

3. RESULTS AND DISCUSSION

For the calibration of the code the cubic cavity problem is solved using the first and the second order accurate schemes. The grid used is fairly coarse, 11x11x11. The first order scheme, with second order artificial dissipation, gives low velocity gradients in the vicinity of the walls as seen in Fig.1 a and b. The second order accurate scheme, on the other hand, predicts the velocity profiles, even with a coarse grid, in agreement with the results given with spectral methods[2].

Shown in Fig.1 c and d is the symmetry plane velocity vectors obtained with the first and second order schemes respectively. The flow Reynolds number is 1000 and the dimensionless time level is 30, the steady state is practically reached.



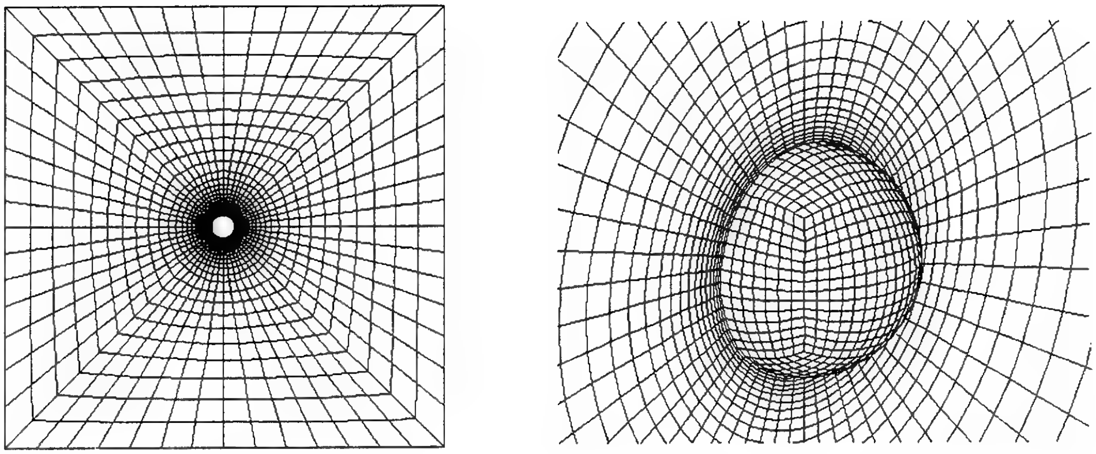
a) vertical centerline.

b) horizontal centerline.

c) Solution with second order dissipation,

d) Solution with fourth order dissipation.

Fig.1 Cubic cavity velocity profiles for Re=1000 in comparison with the results of Ref[2] on the symmetry plane at steady state (a-b). Present solutions with fourth and second order artificial dissipations are shown. Flow velocity vectors on the symmetry plane(c-d). 11x11x11 stretched grid.



a) Full grid.

b) Near body detail.

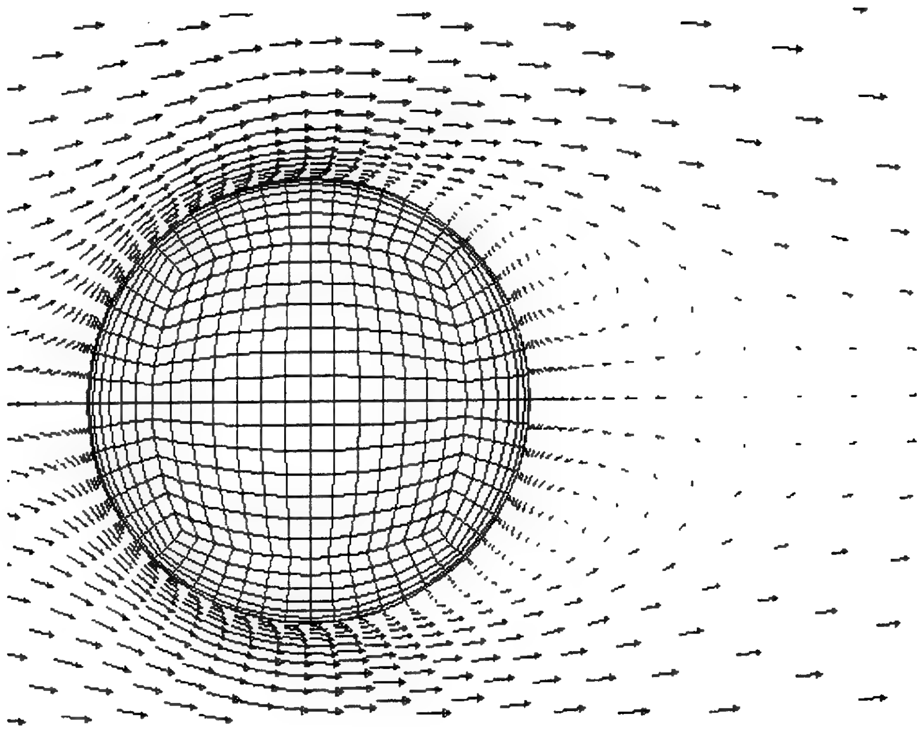
Fig.2 Numerical grid on the symmetry plane of the sphere.

As the second study, laminar flow past a sphere at a high Reynolds number, Re = 162 000, is solved with both the first and the second order schemes. The full grid around the sphere consists of 19127 points and 17640 brick elements as shown in Fig.2a. Fig.2b shows the details of the grid around the body. In Fig.3 a and b the velocity vectors at the symmetry plane at about time level of 4 is plotted. The length of the separation bubble predicted with the both approach is almost the same, however, the width differs significantly. The second order accurate scheme predicts the separation angle close to the value given in [9]. As seen in Fig.3 a and b, the flow is symmetric with respect to the mid plane and, at the upper half of the plane there is a major clockwise recirculating bubble. The details in the separation region, however, is predicted with the accurate scheme as seen in Fig. 3.b, wherein a smaller bubble with clockwise rotation is present at the upstream of the major one. The more detailed picture of right after the shoulder is given in Fig.4 b, where there is a very small counterclockwise rotating bubble in between the major and the minor clockwise rotating bubbles. All these details are smeared out with the first order method as seen in Fig.4 a. even with finer resolution in radial direction.

The third problem solved is related to an institutional project for developing a generic helicopter. The grid around the fuselage is shown in Fig.5, where 11280 brick elements with 12915 nodes are used to resolve the symmetric half of the flow field. The flow Reynolds number, based on the height of the body taken as a characteristic length is 50 000. Shown in Fig.6 a and b is the symmetry plane velocity vector fields at about the steady state obtained with the first and second order schemes, respectively. The results of the second order scheme indicate a longer separation region in the wake, Fig.6 b. A detailed picture of the wake is depicted in Fig.7 a and b, wherein the separation bubble obtained with second order scheme is twice longer than the buble obtained with the first order scheme. Also seen in Fig.6 b is a small separation region at the bottom of the fusalage where there is an unfavorable pressure gradient. The first order scheme can not predict that separation region because of high artificial diffusion. The detailed picture of this unfavorable pressure region is provided in Fig.8 a and b for the first and second order schemes, respectively.

The presssure distribution on the body surface at the symmetry plane is given in Fig.9. According to this figure the pressure values follow, in general, the same trend for the both solutions, however the unfavorable pressure gradient region at the bottom surface indicate where the two solutions do not agree.

a)Second order dissipation



b)Fourth order dissipation

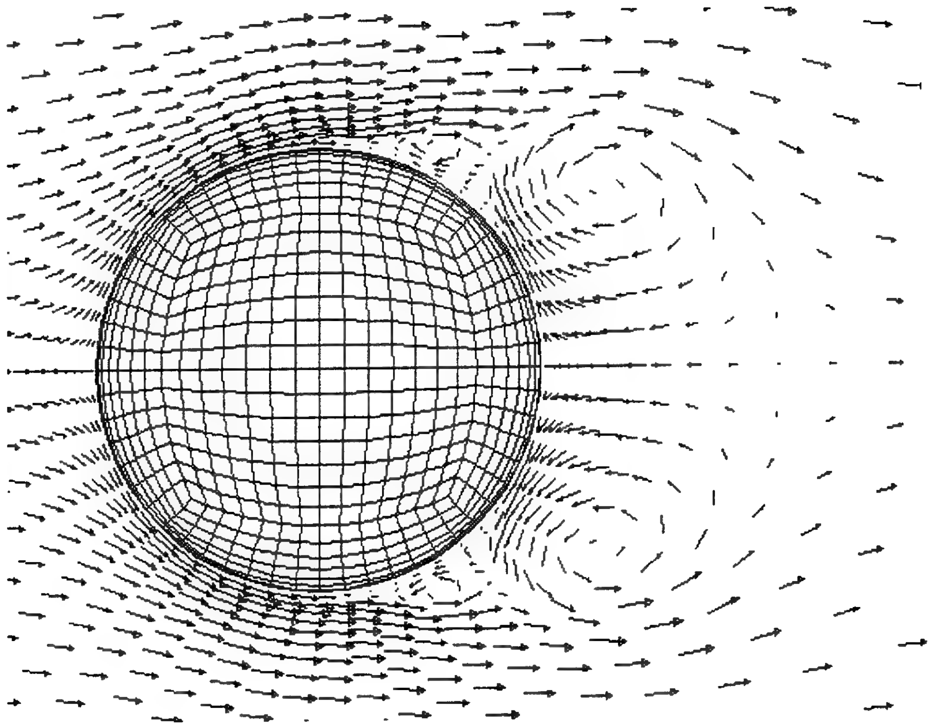
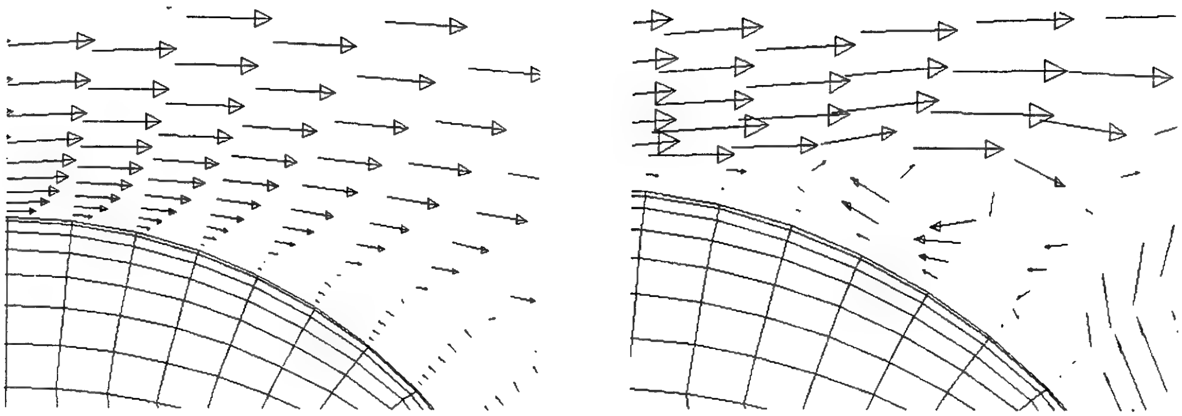


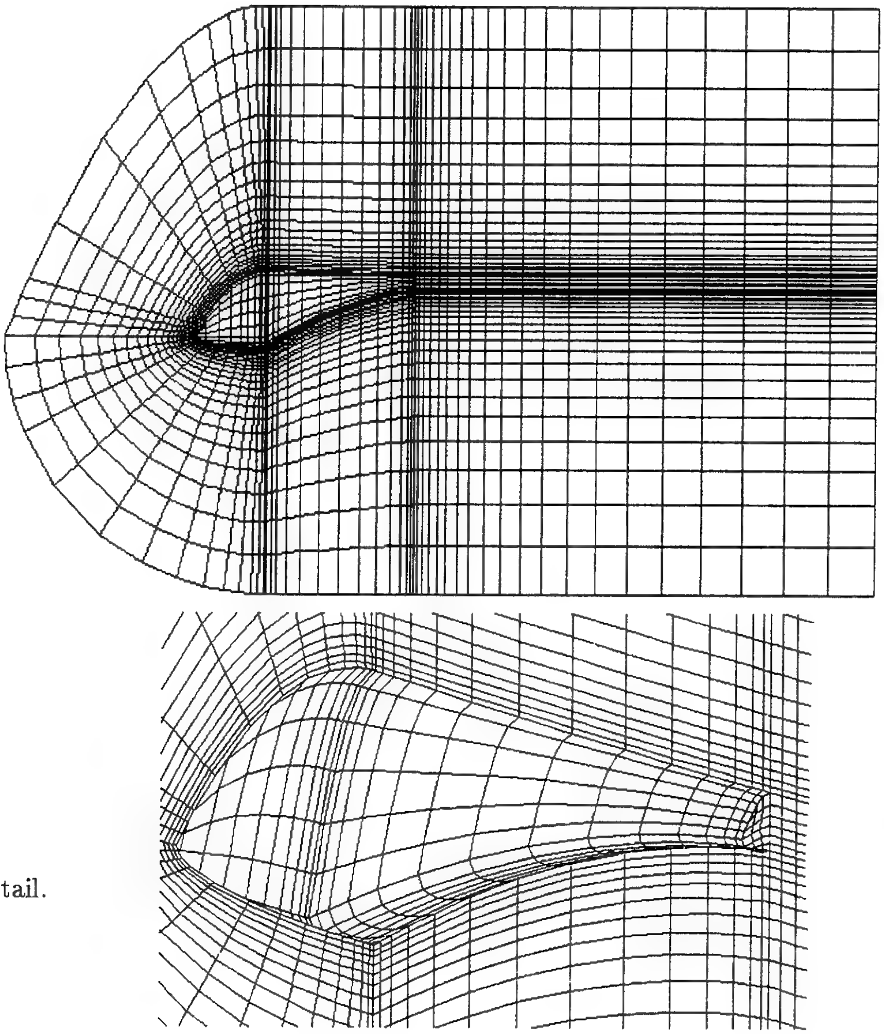
Fig.3 Velocity vectors on the symmetry plane of the sphere, Re=162 000.



a) Solution with second order dissipation.

b) Solution with fourth order dissipation.

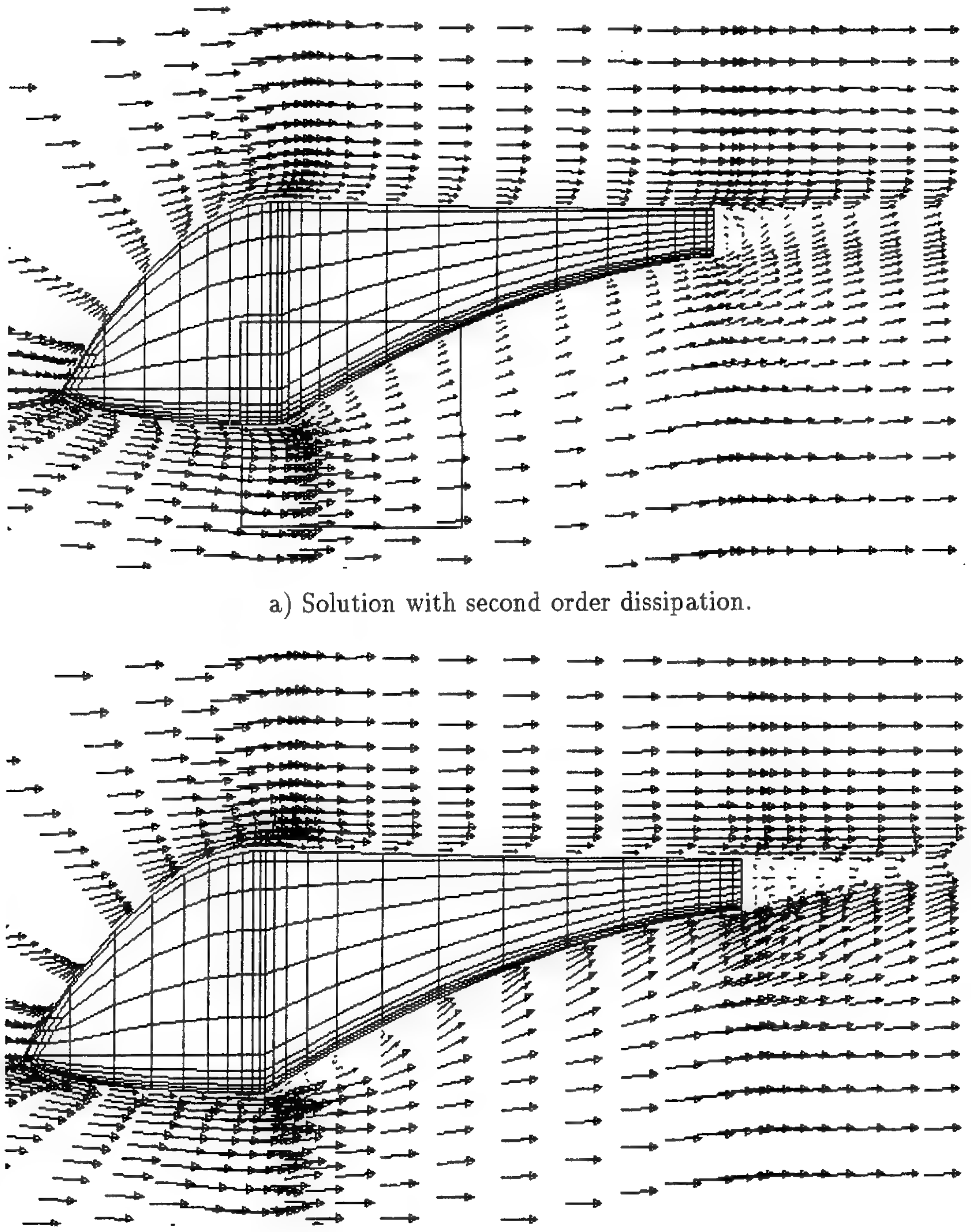
Fig.4 Velocity vector details at the symmetry plane on the right shoulder of the sphere.



a) Full grid.

b) Near body detail.

Fig.5 Numerical grid on the symmetry plane of the helicopter fuselage.



a) Solution with second order dissipation.

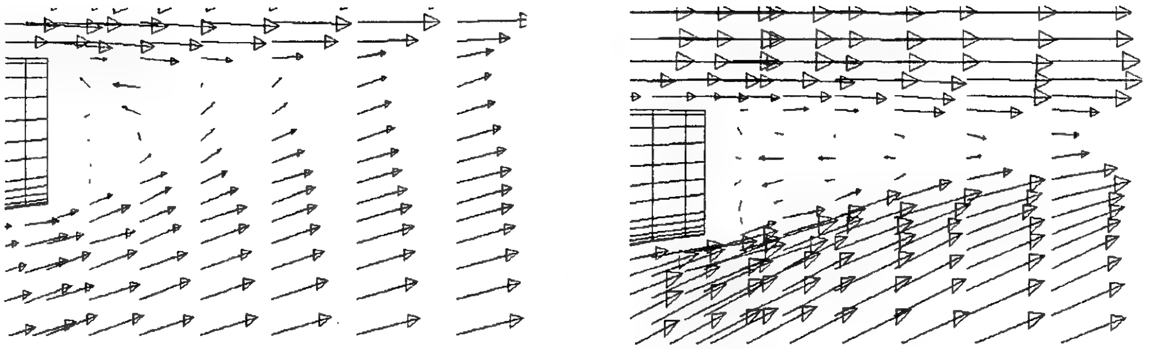
b) Solution with fourth order dissipation.

Fig.6 Velocity vectors on the symmetry plane of the fuselage, Re=50 000.

Shown in Table 1 is the drag coefficient values for the sphere compared with the experimental data[9]. As seen from the values, the first order scheme over estimates the coefficient values whereas the second order scheme under estimates them as compared to experimental values. The drag coefficient values evaluated for the helicopter fuselage with both schemes are also given in Table 1.

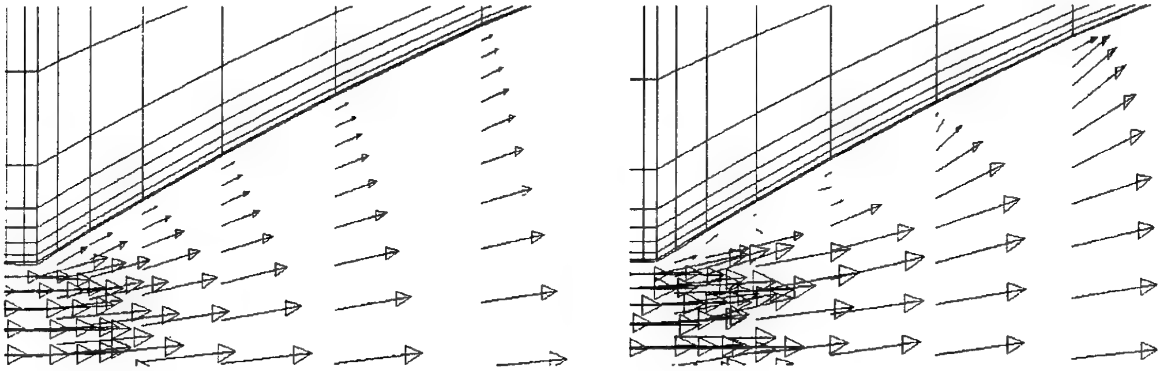
CONCLUSION

A computer code based on a second order accu-



a) Solution with second order dissipation, b) Solution with fourth order dissipation.

Fig.7 Velocity vector details at the wake region of the fuselage, Re=50 000.



a) Solution with second order dissipation, b) Solution with fourth order dissipation.

Fig.8 Velocity vector details at the unfavorable pressure region of the fuselage.

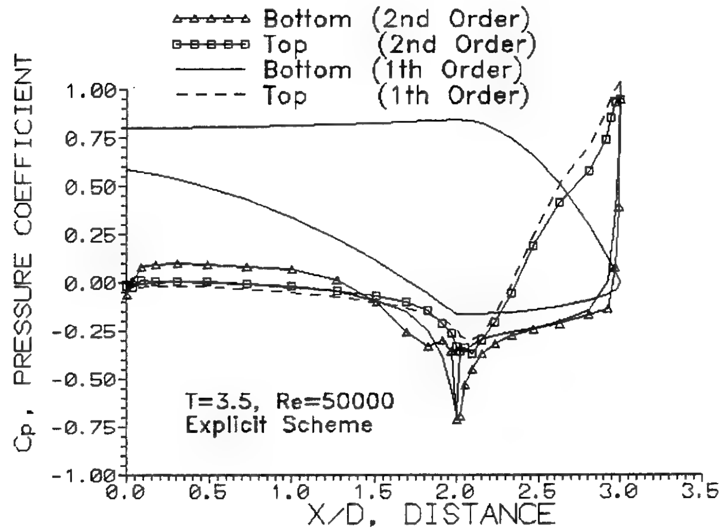


Fig.9 Pressure coefficient (Cp) values on the body surface at the symmetry plane.

Geometry	**Re number**	Scheme	C_d
Sphere	162 000	Experiment [9]	0.47
		First order	0.52
		Second order	0.38
Fuselage	50 000	First order	0.20
		Second order	0.11

Table.1 Drag coefficient values for the sphere and the helicopter fuselage.

rate scheme is developed and implemented for flows involving large separations and strong recirculations about arbitrary shapes.

The results obtained for various test case are in good agreement with the existing numerical and experimental data.

The code is implemented satisfactorily to predict the drag coefficient of a generic helicopter fuselage.

ACKNOWLEDGEMENT

This work is partially supported by ITU, Research Fund under project number 494.

REFERENCES

[1] Ü. Gülçat, *An Explicit FEM for 3-D General Viscous Flow Studies Based on E-B-E Solution Algorithms*, Comp. Fluid Dyn., vol.4, pp.73-85, 1995.

[2] Hwar C.Ku, Richard S. Hirsh and Thomas D. Taylor, " A Pseudospectral Method for Solution of the Three-Dimensional Incompressible Navier-Stokes Equations", Journal of Computational Physics, 70, pp.439-462, 1987.

[3] A. Mizukami and M. Tsuchiya, *A Finite Element Method for the Three-Dimensional Non-Steady Navier-Stokes Equations*, Int.J.Num Meth.in Fluids, vol.4, pp.349-357, 1984.

[4] O.C. Zienkiewicz and R.L. Taylor, *The Finite element method*, volume 1, McGraw-Hill book company, London, 1989.

[5] A.R. Aslan, F.O. Edis, Ü. Gülçat and E. Gürgey, *Prediction of General Viscous Flows Using a Finite Element Method*, Proceedings of the 8th International Conference on Numerical Methods in Laminar and Turbulent Flows, Swansea, U.K., July 18-23, 1993.

[6] M.F. Webster and P. Townsend, *Development of a Transient Approach to Simulate Newtonian and Non-Newtonian Flow*. NUMETA'90, Numerical Methods in Engineering: Theory and Applications, Edts: G.N. Pande and J. Middleton, Vol.2, Elsevier Publications, U.K., pp.1003-1012, 1992.

[7] Ü. Gülçat, A.R. Aslan and A. Mısırlıoğlu, *Aerodynamics of Fuselage and Store-carriage Interaction using CFD*, Agard 76th Fluid Dynamics Panel Symposium on Aerodynamics of Store Intergration, 24-27 April, Ankara, Turkey, 1995.

[8] Y. Kallinderis and K. Nakajima, *Finite Element Method for Incompressible Viscous Flows with Adaptive Hybrid Grids*, AIAA Journal, vol.32, No.8, pp.1617-1625, August, 1994.

[9] S.W. Churchill, *Viscous flows, the practical use of theory*, pp.360-400, Butterworths series in chemical engineering, Butterworth publishers, USA, 1972.

Convergence Acceleration of the Navier-Stokes Equations through Time-Derivative Preconditioning

Charles L. Merkle, Sankaran Venkateswaran, and Manish Deshpande
Propulsion Engineering Research Center
The Pennsylvania State University
University Park, Pa 16802

SUMMARY

Chorin's method of artificial compressibility is extended to both compressible and incompressible fluids by using physical arguments to define artificial fluid properties that make up a local preconditioning matrix. In particular, perturbation expansions are used to provide appropriate temporal derivatives for the equations of motion at both low speeds and low Reynolds numbers. These limiting forms are then combined into a single function that smoothly merges into the physical time derivatives at high speeds so that the equations are left unchanged at transonic, high Reynolds number conditions. The effectiveness of the resulting preconditioning procedure for the Navier-Stokes equations is demonstrated for wide speed and Reynolds number ranges by means of stability results and computational solutions. Nevertheless, the preconditioned equations sometimes fail to provide a solution for applications for which the non-preconditioned equations converge. Often this is because the reduced dissipation in the preconditioned equations results in an unsteady solution while the more dissipative non-preconditioned equations result in a steady state. Problems of this type represent a computational challenge: it is important to distinguish between non-convergence of algorithms, and the non-existence of steady state solutions.

1 INTRODUCTION

Time-marching techniques have proven to be very effective for the computation of high Reynolds number flows in the transonic, supersonic and hypersonic regimes. These methods, however, become inefficient at low speed or low Reynolds number conditions including the near wall regions of high Reynolds number flows. For this reason, incompressible and low speed computations were dominated by pressure-based procedures [1] for many years. Chorin's pseudo-compressibility method [2], which has become widely accepted for incompressible flows [3], opened one avenue for applying time-marching procedures to incompressible flows but there was little realization that this procedure could be broadened to enable computations at all speeds until recently.

Extensions of time-marching methods to low Mach number compressible flows became possible with the realization that it was the stiffness of the eigenvalues that slowed convergence at low speeds. Low Mach number perturbation procedures were first used to remove these problems [4] and were used in pressure-based methods to compute low speed compressible solutions. The implementation of time-marching methods to the low Mach number perturbation equations were first reported by Gustafsson [5], followed by extensive applications by the present authors [6]. Perturbation expansion methods have also been extended to combustion problems [7]. Of these perturbation expansion methods, some (6, 7] used the more conventional expansion procedures based on the square of the Mach number, while others [5,8] expanded the equations in terms of the first power of the Mach number.

In parallel with these perturbation procedures, local preconditioning methods in which the time derivatives of the equations of motion are multiplied by a matrix to control the eigenvalues have also been used to enhance convergence [8-16]. Unlike the perturbation equations, these preconditioned equations are valid at all speeds, and so have a potential for providing uniform convergence over all Reynolds and Mach number regimes. Two distinct philosophies have been followed in developing these preconditioning methods. One uses the perturbation procedures described above and deals with the full Navier-Stokes equations and includes the Euler equations as a special case [11-14]. The intent of this approach is to improve convergence at low speeds and Reynolds numbers only, while leaving it unaltered at high Reynolds numbers and high speeds (transonic and above) where it is already quite efficient. This method has been applied extensively to a wide variety of practical applications.

The second approach [15, 16] provides a rigorous method for developing a preconditioning matrix for the Euler equations, but equally rigorous extension to the Navier-Stokes equations appears doubtful. This preconditioning procedure is intended to provide optimum convergence over the entire Mach number regime, but limited applications have thus far demonstrated convergence enhancement only in the low Mach number regime [16]. Even there, this second method is generally less effective than that provided by the perturbation-expansion-based methods. Further, the convergence enhancement to be had at transonic and supersonic speeds is very limited because time-marching methods are already efficient there so that substantial improvements are unlikely.

The purpose of the present paper is to demonstrate how a viscous preconditioning procedure can be developed from the basic physics of the flow using low speed and low Reynolds number perturbation expansions. As a part of this development, the link between our compressible preconditioning method and the artificial compressibility method of Chorin is shown. Following some representative examples of convergence enhancement for a wide variety of problems, the paper closes by addressing the issue of the robustness of preconditioning methods. One specific example is given in which the preconditioned methods fail to provide convergence to a steady state. Detailed investigation shows that the physical problem is unsteady and a steady solution fails to exist. The reduced artificial dissipation in the preconditioned solution makes this unsteadiness more apparent. The prospect of distinguishing non-convergence from the non-existence of steady state solutions is thus raised as a challenge facing CFD techniques.

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

2 THE EQUATIONS OF MOTION

The equations of motion can be written in conservative form as:

$$\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}+\frac{\partial G}{\partial z} = L_v(Q_v) \tag{1}$$

where the viscous terms are given by the operator L_V, and the vectors Q, Q_V, and E are

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{pmatrix} \quad Q_v = \begin{pmatrix} p \\ u \\ v \\ w \\ T \end{pmatrix} \quad E = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho u h^o \end{pmatrix} \tag{2}$$

with an analogous definition for F and G. Here, ρ represents the density, p is the pressure, and u, v and w are the Cartesian velocity components in the x, y and z directions respectively. The total energy, e, is the sum of the internal energy, ε, and the kinetic energy,

$$e = \rho\varepsilon + \frac{1}{2}\left(u^2+v^2+w^2\right) \tag{3}$$

The enthalpy, h, is related to the internal energy and the pressure,

$$\rho h = \rho\varepsilon + p \tag{4}$$

and for a perfect gas can be expressed as a function of the temperature alone, $h = h(T)$. The stagnation enthalpy is defined as $h^o = h + (u^2 + v^2 + w^2)/2$. The formulation is completed by the perfect gas equation of state which we write as,

$$\rho = \rho(p,T) = p/RT \tag{5}$$

to emphasize that the density depends on the temperature and pressure. This form makes it possible to include incompressible fluids and perfect gases in a single procedure.

The "viscous" vector, Q_V, that appears in Eqs. 1 and 2 represents the dependent variables that appear naturally in the diffusion terms. Because the first cell of this variable (corresponding to the continuity equation) is null, we choose to fill it with the pressure, p. This choice makes Q_V a unique function of the conservative variable Q . For convenience, we use this set of primitive variables as our primary dependent variable set, but we retain the conservative fluxes.

The variables in the time derivative can easily be changed from Q to Q_V by means of the chain rule,

$$\left(\frac{\partial Q}{\partial Q_v}\right)\frac{\partial Q_v}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}+\frac{\partial G}{\partial z}=L_v(Q_v) \tag{6}$$

where $\partial Q/\partial Q_V$ represents the Jacobian,

$$\frac{\partial Q}{\partial Q_v} = \begin{pmatrix} \rho_p & 0 & 0 & 0 & \rho_T \\ u\rho_p & \rho & 0 & 0 & u\rho_T \\ v\rho_p & 0 & \rho & 0 & v\rho_T \\ w\rho_p & 0 & 0 & \rho & w\rho_T \\ h^o\rho_p - 1 & \rho u & \rho v & \rho w & h^o\rho_T + \rho h_T \end{pmatrix} \tag{7}$$

where ρ_p, ρ_T, and h_T are partial derivatives. For a perfect gas $\rho_T = -\rho/T$; $\rho_p = 1/RT$, $h_T = \gamma R/(\gamma - 1)$ where γ is the ratio of specific heats. Note that h_T is the specific heat at constant pressure.

Other matrices of interest include the Jacobian, $A_V = \partial E/\partial Q_V$,

$$A_v = \begin{pmatrix} u\rho_p & \rho & 0 & 0 & u\rho_T \\ u^2\rho_p + 1 & 2\rho u & 0 & 0 & u^2\rho_T \\ uv\rho_p & \rho v & \rho u & 0 & uv\rho_T \\ uw\rho_p & \rho w & 0 & \rho u & uw\rho_T \\ h^o\rho_p & \rho\left(h^o + u^2\right) & \rho uv & \rho uw & u\left(h^o\rho_T + \rho h_T\right) \end{pmatrix} \tag{8}$$

with analogous expressions for $B_v = \partial F/\partial Q_v$ and $C_v = \partial G/\partial Q_v$.

3 LOW MACH NUMBER SCALING

The eigenvalues of (6) determine the convergence rate of the time-marching algorithm. These eigenvalues are obtained from the roots of the fifth order polynomial:

$$\left(\frac{\partial Q_v}{\partial Q}A_v - \lambda I\right) = 0 \tag{9}$$

which are readily found to be u, u, u, $u \pm c$ where the acoustic speed, c , is given by,

$$c^2 = \frac{\rho h_T}{\rho_T + \rho\rho_p h_T} \tag{10}$$

The speed of sound reduces to the familiar relation, $c^2 = \gamma RT$, for a perfect gas, while, for an incompressible fluid where $\rho_p = \rho_T = 0$, and the speed of sound becomes infinite causing the time derivatives in the continuity equation to vanish so that continuity reduces to $\nabla \bullet \mathbf{V} = 0$.

To ensure uniform, efficient convergence over all speed ranges, we replace the matrix $(\partial Q/\partial Q_V)$ in (9) by a preconditioning matrix, Γ_v, and consider the solution of the modified equation,

$$\Gamma_v\frac{\partial Q_v}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}\frac{\partial G}{\partial z}=L_v(Q_v) \tag{11}$$

We define Γ_v in a form analogous to $\partial Q/\partial Q_V$, by replacing the fluid properties, ρ_p, ρ_T and h_T, by the artificial quantities, ρ'_p, ρ'_T and h'_T respectively. These quantities represent a three-parameter preconditioning system whose values can be chosen to ensure well-conditioned eigenvalues at all speeds, thereby ensuring fast, efficient convergence. The definition of the three parameters in the preconditioning matrix, Γ_v, will be obtained from perturbation analyses of the equations of motion at low speeds and low Reynolds numbers. Their presence introduces an artificial speed of sound, c', that ensures that eigenvalue stiffness is avoided. Additional restrictions on the preconditioning matrix have been given by Viviand [9] and Choi and Merkle [12].

To overcome the difficulties at low speeds, we expand Q_v in the power series,

$$Q_v = Q_{v0} + \varepsilon Q_{v1} + \ldots \qquad (12)$$

where $\varepsilon = M^2$. Upon substituting this expression into (11), we obtain to order $1/\varepsilon$, p_o = constant, which says that the thermodynamic pressure is externally imposed. Scaling the temporal and spatial derivatives of pressure to order unity then causes the term, p_1, to appear in the zeroeth order equations. To reflect this, we define the vector, Q'_{v0},

$$Q'_{v0} = \left(p_1, u_0, v_0, w_0, T_0\right)^T \qquad (13)$$

so that the equation system that is valid to order unity becomes (for simplicity here, we write only the one-dimensional equations),

$$\Gamma_{v0}\frac{\partial Q'_{v0}}{\partial t} + A_{v0}\frac{\partial Q'_{v0}}{\partial x} = L_{v0}\left(Q'_{v0}\right) \qquad (14)$$

The corresponding matrices Γ_{v0} and A_{VO} are given by evaluating Γ_v and A_V with the values Q_{V0}.

Requiring that the temporal pressure derivative be retained in the continuity equation in the low speed limit implies that ρ_p must be of order one, or that ρ'_p, is given by

$$\rho'_p = k_p/V_r^2 \qquad (15)$$

where k_p is a constant of order unity and V_r is an appropriate reference velocity.

To ensure that the energy equation is uncoupled from the continuity and momentum equations in the incompressible limit, we make the variable ρ'_T proportional to ρ_T,

$$\rho'_T = k_T\rho_T \qquad (16)$$

where k_T is a quantity whose value is less than or equal to one. This replacement causes ρ'_T to vanish when ρ_T goes to zero. Specific values for k_T are 0 or 1.

Placing Eqs. 15 and 16 in Γ_v, gives well conditioned eigenvalues in the limit of low Mach numbers,

$$\begin{pmatrix} k_p/V_r^2 & 0 & 0 & 0 & k_T\rho_T \\ u\,k_p/V_r^2 & \rho & 0 & 0 & k_T u\rho_T \\ v\,k_p/V_r^2 & 0 & \rho & 0 & k_T v\rho_T \\ w\,k_p/V_r^2 & 0 & 0 & \rho & k_T w\rho_T \\ h^o\,k_p/V_r^2 - 1 & \rho u & \rho v & \rho w & k_T h^o\rho_T + \rho h'_T \end{pmatrix} \qquad (17)$$

The quantity h'_T is, as yet, free.

With the special values for ρ'_p and ρ'_T given in (15) and (16), two eigenvalues of $\Gamma_v^{-1}A_v$ become equal to the particle speed u.. The third eigenvalue also equals u. if $h_T = h'_T$ or if the physical properties ρ_p and ρ_T are zero as in incompressible flow. For these conditions, the full set of eigenvalues of $\Gamma_v^{-1}A_v$ is:

$$\Lambda_p = diag\left[u, u, u, u\left(a+\sqrt{b}\right), u\left(a-\sqrt{b}\right)\right] \qquad (18)$$

where the quantities a and b are given by,

$$a = \left(\frac{1}{2d}\right)\left[\rho\left(\rho_p h'_T + \rho'_{ph_T}\right) + \rho_T + \rho'_T\right] \qquad (19a)$$

$$\begin{aligned} b = \left(\frac{1}{2d}\right)^2 \Bigg[& \frac{4\rho h'_T d}{u^2} + \rho^2\left(\rho_p h'_T - \rho'_p h_T\right)^2 \\ & + \left(\rho_T - \rho'_T\right)^2 + 2\rho\Big\{\Big[\rho'_T\left(\rho_p h'_T - \rho'_p h_T\right) \\ & - 2\rho'_p\left(\rho_T h'_T - \rho'_T h_T\right)\Big] + \Big(2\rho_p\left(\rho_T h'_T - \rho'_T h_T\right) \\ & - \rho_T\left(\rho_p h'_T - \rho'_p h_T\right)\Big)\Big\}\Bigg] \end{aligned} \qquad (19b)$$

Inspection of the generalized acoustic eigenvalues in (19) shows that the physical properties, ρ_p and ρ_T that cause the speed of sound in incompressible flows to be infinite no longer appear in the denominator; only the artificial properties ρ'_p and ρ'_T do. Replacing these physical properties by properly defined artificial properties alleviates the decoupling between the pressure and momentum terms in incompressible flows, and makes time-marching practical for both incompressible and low speed compressible flow computations. For incompressible flows, this approach leads to the artificial-compressibility method of Chorin as is shown below.

Replacing ρ'_p by k_p/V_r^2 and ρ'_T by $k_T\rho_T$ as suggested by the low-Mach number scaling provides eigenvalues that are well -conditioned for low speeds. For a perfect gas, the coefficients, a and b, become:

$$a = \frac{\rho}{2d}\left[\frac{c_p k_p}{V_r^2}\left(1 + \frac{\gamma M r^2}{k_p}\frac{h'_T}{c_p}\right) - \left(\frac{1+k_T}{T}\right)\right] \qquad (20a)$$

$$\begin{aligned} b = \left(\frac{\rho}{2d}\right)^2 \Bigg[& \frac{4h'_T d}{\rho u^2} + \left(\frac{c_p k_p}{V_r^2}\right)^2\left(1 - \frac{\gamma M r^2}{V_r^2}\frac{h'_T}{c_p}\right) + \left(\frac{1-k_T}{T}\right)^2 \\ & + \frac{2c_p k_p}{V_r^2 T}\left\{\left(1 - \frac{\gamma M r^2}{V_r^2}\frac{h'_T}{c_p}\right)\left(1 - h'_p\right) + 2\left(\frac{h'_T}{c_p} - k_T\right)\left(1 - \frac{\gamma M r^2}{V_r^2}\right)\right\}\Bigg] \end{aligned} \qquad (20b)$$

where $M_r = V_r/c$ is the reference Mach number. The behavior of this eigenvalue is difficult to determine from the algebraic form, but it is seen that as the Mach number goes to zero, the eigenvalues approach a constant times the particle speed. Numerical checks verify that this constant is of order unity and the eigenvalues are well-conditioned. Stability results for this condition are presented later.

For incompressible flow, the coefficients *a* and *b* become:

$$a = 1/2, \quad b = \left(V_r^2 / k_p u^2 + 1/4\right) \tag{21}$$

which are clearly well-behaved. Choosing $k_p = 2$ gives eigenvalues whose ratio is no worse than 2. This choice is identical to the artificial compressibility method of Chorin [2]. We also note that for incompressible flow, it is not necessary to set $h_T = h'_T$ to obtain simple algebraic eigenvalues, and the third "particle" eigenvalue becomes $\lambda = uh_T / h'_T$, so the parameter h'_T can be selected to control convergence in the energy equation if desired.

4 LOW REYNOLDS NUMBER EQUATIONS

Having obtained some understanding of the way the Euler equations scale, we now turn to the Navier-Stokes equations and consider their proper scaling in the limit of low Reynolds numbers. Here, we use a similar perturbation expansion, but we let the small parameter, ε, be the Reynolds number, *Re*. We begin by premultiplying (11) by the matrix P_v^{-1},

$$P_v^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 & 0 \\ -v & 0 & 1 & 0 & 0 \\ -w & 0 & 0 & 1 & 0 \\ -h+\frac{1}{2}V^2 & -u & -v & -w & 1 \end{pmatrix} \tag{22}$$

This multiplication gives the matrix

$$P_v^{-1}\Gamma_v = \begin{pmatrix} \rho'_p & 0 & 0 & 0 & \rho'_T \\ 0 & \rho & 0 & 0 & 0 \\ 0 & 0 & \rho & 0 & 0 \\ 0 & 0 & 0 & \rho & 0 \\ -1 & 0 & 0 & 0 & \rho h'_T \end{pmatrix} \tag{23}$$

and results in the convective terms,

$$\left(\frac{\partial \rho u}{\partial x}, \rho u \frac{\partial u}{\partial x} + \frac{\partial p}{\partial x}, \rho u \frac{\partial w}{\partial x}, \rho u \frac{\partial h}{\partial x} - u \frac{\partial p}{\partial x}\right)^T \tag{24}$$

for the x-direction. Multiplication by P_v^{-1} does not affect the viscous terms in the momentum equations, but the corresponding terms in the energy equation reduce to the conduction term plus the viscous dissipation. The modified energy equation becomes:

$$-h'_p \frac{\partial p}{\partial t} + \rho h'_T \frac{\partial T}{\partial t} - u \frac{\partial p}{\partial x} + \rho u \frac{\partial h}{\partial x} = \nabla \cdot k \nabla T + \Phi \tag{25}$$

where Φ is the viscous dissipation, and we have omitted convective terms in *y* and *z*. These equations can then be scaled for low Reynolds numbers to see how our three parameters ρ'_p, ρ'_T and h'_T must behave in the diffusion-dominated limit.

For low Reynolds numbers, we scale the momentum equations such that the temporal derivatives and the pressure gradient remain of the same order as the viscous terms as the Reynolds number goes to zero. This defines the proper scaling for the pressure $(p_r = \mu_r V_r/L)$ and the time $(t_r = \rho_r L^2 / \mu_r)$, but imposes no conditions on any of our three preconditioning parameters.

Using this reference pressure and time and requiring that the temporal term in the continuity equation balance the convective terms at low Reynolds numbers, results in the condition on ρ'_p,

$$\rho'_p = k'_p Re^2 / V_r^2 \tag{26}$$

To prevent the temporal derivative of the temperature from appearing in the momentum equation, we also require,

$$\rho'_T = k'_T \rho_T Re / T_r \tag{27}$$

In these expressions, k_p is a constant of order unity, while k_T is less than or equal to one.

Scaling the energy equation in a manner consistent with these definitions results in the requirement,

$$h'_T = k'_h c_{pr} / Pr \tag{28}$$

where c_{pr} is a reference specific heat, Pr is a reference Prandtl number, and k'_h is another constant of order one.

The resulting low Reynolds number equations in one-dimension then become:

$$k'_p \frac{\partial p}{\partial t} + \rho \frac{\partial u}{\partial x} + \rho_p u \frac{\partial p}{\partial x} + \rho T u \frac{\partial T}{\partial x} = 0$$

$$\rho \frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = \frac{\partial}{\partial x} \frac{4}{3} \mu \frac{\partial u}{\partial x} \tag{29}$$

$$k'_h \frac{\partial T}{\partial t} = \nabla \cdot k \nabla T + \Phi$$

Note that in the energy equation, we have assumed that the quantity $V_r^2 / c_{pr} T_r$ is small. Retaining it adds a pressure gradient term to the energy equation, but does not affect the requirements placed upon our three parameters. Equations (30) are the creeping flow equations.

5 SUMMARY OF PRECONDITIONING PROCEDURE

The correct asymptotic form of the three parameters, ρ'_p, ρ'_T and h'_T, as determined from the low Mach number scaling and low Reynolds number scaling is summarized in Table I. Use of these values ensures that the equations are properly scaled in these two limits. To use these parameters for computations at other than the limiting transitions smoothly between these limits while also approaching the non-preconditioned equations for high Reynolds number, transonic flows. This functional form will be developed by combining these limiting values into a single continuous function and then verifying the results first by means of stability theory, then by simplified computational problems, and finally by practical applications at low speed, low Reynolds number and transonic conditions.

Preconditioning the Euler equations is relatively easy, but preconditioning the Navier-Stokes equations is more difficult for several reasons. First of all, the appropriate Reynolds number must be determined. Stability results show that the cell Reynolds number, $u\Delta x/\nu$ (where u represents the local velocity, Δx represents the grid spacing and ν is the kinematic viscosity), is the appropriate viscous scale, and that diffusive effects become dominant at cell Reynolds numbers less than unity. The transition from inviscid- to viscous-dominated flows thus depends on both the flowfield and the grid. Viscous flows can switch from convection-dominated to diffusion-dominated because of increased grid resolution or stretching. The second reason for difficulty arises because the presence of boundary layers at high Reynolds numbers requires high aspect ratio grids with fine resolution normal to the walls. Correspondingly, there are two cell Reynolds numbers of widely differing magnitude. The one based on the normal grid spacing is generally diffusion dominated, while the one based on the streamwise spacing is generally convection-dominated. The issue in viscous preconditioning is to deal with near-wall cells that are viscously-dominated in one direction and convectively-dominated in the other, while simultaneously treating convectively-dominated cells in regions away from the walls.

We demonstrate two ways in which the limiting forms of the artificial properties in Table I can be combined into a single function that can be used over the full Reynolds-Mach number domain. The parameter ρ'_p is the primary quantity in controlling eigenvalues, and we begin by considering this quantity. The simplest procedure is to choose ρ'_p as the minimum of the viscous and inviscid values,

$$\rho'_p = \left(k_p / V_r^2\right) Min\{1, \mathbf{Re}\} \qquad (30)$$

where we have used the same constant at both conditions. This is equivalent to using the smaller of an inviscid or viscous time step and has proven effective in many problems [11]. Clearly, this corresponds to switching from the inviscid to the viscous value when the Reynolds number goes below unity. The most appropriate Reynolds number for this switch is the cell Reynolds number.

Table I. Preconditioning Parameters Dictated by Reynolds and Mach Number Scaling

Term	Low Mach Number	Low Reynolds Number
ρ'_p	k_p/V_r^2	$k'_p\,\mathbf{Re}^2/V_r^2$
ρ'_T	$k_T\rho_T$	$k'_T\,\rho_T\,Re$
h'_T	$k_h h_T$	$k'_h\,h_T/Pr$

The function in Eq. 30 can likewise be made to merge smoothly with the physical properties at transonic conditions by noting that at Mach one, $V_r^2 = c^2$, so that if we choose $k_p = k'_p = \gamma$, Eq. 30 degenerates to $\rho'_p = \rho_p$ at Mach one. (In computations for incompressible flow we have generally chosen $k_p = k'_p = 1.33$). The remaining artificial properties can be made continuous by setting $k_T = k'_T = 0$, and by setting $k_h = 1$ and $k'_h = \mathrm{Pr}$. This latter choice does not precisely satisfy the viscous matching condition, but since the Prandtl number for most gases is near one, it is close enough to give good results. All the examples we give are based on this combination of artificial properties.

The second procedure is similar, but instead of using a function with a discontinuous slope for ρ'_p we make both the function and its derivatives continuous. Here we define the three parameters as:

$$\begin{aligned} \rho'_p &= \frac{\gamma}{V_r^2}\left(\frac{\mathbf{Re}^2}{1+\mathbf{Re}^2}\right) \\ \rho'_T &= \rho_T\left(\frac{\mathbf{Re}}{1+\mathbf{Re}}\right) \\ h'_T &= \frac{h_T}{\mathbf{Pr}}\left(\frac{1+\mathbf{Re}\,\mathbf{Pr}}{1+\mathbf{Re}}\right) \end{aligned} \qquad (31)$$

These functions reach the proper limits at low Reynolds numbers, low Mach numbers, and at high Reynolds number, transonic conditions. In particular, the function for h'_T switches continuously from unity to 1/Pr as the cell Reynolds number goes through unity. When the Mach number approaches unity, Γ_v, approaches the physical Jacobian, $\partial Q/\partial Q_v$, and the preconditioned equations become identically the physical equations. Choosing k_T = 0, as in the first example gives simpler pre-conditioned equations, but only causes the modified eigenvalues of the equations to approach the physical eigenvalues as the Mach number goes to unity. The equations remain distinct.

In summary , we scale the time derivatives at high cell Reynolds numbers to keep the convective eigenvalues well-conditioned, whereas at low cell Reynolds numbers, we scale so that the equations reduce to simple diffusive equations. We also scale the dominant convective speed so that it is the same order as the diffusive time-scale. The low Reynolds number scaling causes the convective terms to become stiff, but because they are small, this doesn't slow convergence. To assess this scaling, we use Fourier stability theory for the full Navier-Stokes equations using Reynolds number and Mach number as parameters.

6 STABILITY AND CONVERGENCE OF THE PRECONDITIONED EQUATIONS

We begin by comparing the stability characteristics of the two-dimensional Euler equations with and without preconditioning at a Mach number of 0.01 and a flow angle

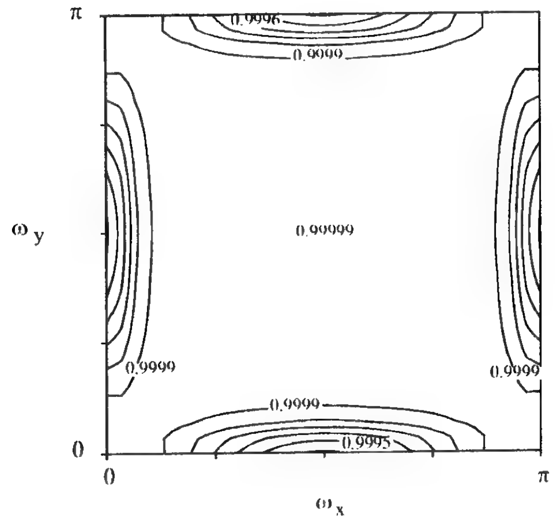


Figure 1: Euler: CD-ADI, M=0.01, No Preconditioning, CFL=5

of 45°. Figure 1 shows results for the non-preconditioned case, while Fig. 2 is for the case with preconditioning. Both of these stability predictions are for central differencing in space and ADI approximate factorization in time. The stability results without preconditioning indicate the amplification factor is nearly unity (0.9999) over the mid- and low-wave-number regions, thereby vividly demonstrating the stiffness that is encountered at low speeds.

By contrast, the amplification factors in Fig. 2 for the preconditioned case are quite reasonable with damping rates of around 0.9 over most of the mid- and low-wave-number ranges with sharp fall-off along the axes (except at the corners) indicating that the preconditioned system will provide fast, efficient convergence at this low Mach number condition. We do note that the amplification factor goes to unity in all four corners, but these peaks are easily removed by a small amount of artificial dissipation. Companion stability results (not shown) indicate that this preconditioned stability result is independent of Mach number, and that it is nearly identical to that for the non-preconditioned equations at a Mach number of 0.7, suggesting that convergence with the preconditioned system will be similar to the efficient convergence observed with the non-preconditioned system at high subsonic Mach numbers. The non-preconditioned eigenvalues in Fig. 1, however, indicate that this case will converge very slowly, an indication that if verified by computations. This demonstrates the ease with which the stiffness in the Euler equations can be removed.

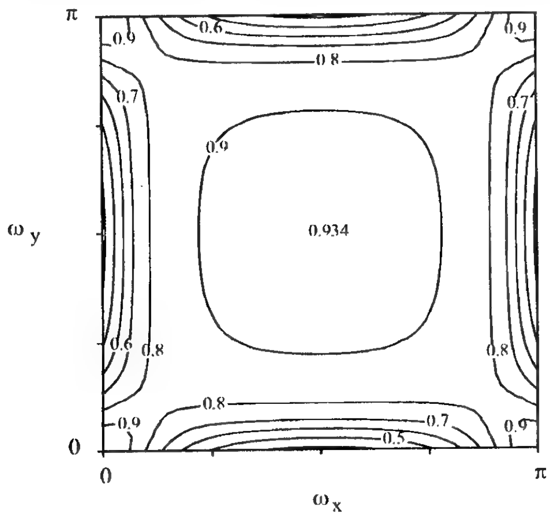


Figure 2: Euler: CD-ADI, M=0.01, With Preconditioning, CFL=5

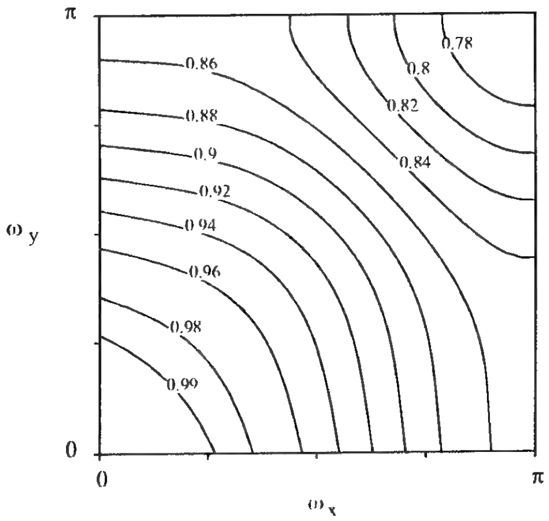


Figure 3: Euler: LGS-4, I/III, M=0.01, No Preconditioning, CFL=20

To further demonstrate the effectiveness of the preconditioning, we show stability results for similar conditions in Figs. 3 and 4, except that upwind differencing is used for the spatial discretization and line Gauss-Seidel approximate factorization is used for the solution procedure. Figure 3 shows the non-preconditioned stability results for $M = 0.01$. These eigenvalues again contain an unacceptable stiffness in the low-wave-number region. This stiffness is, however, removed by the preconditioning as shown in Fig. 4. Again, this preconditioning renders the stability results essentially

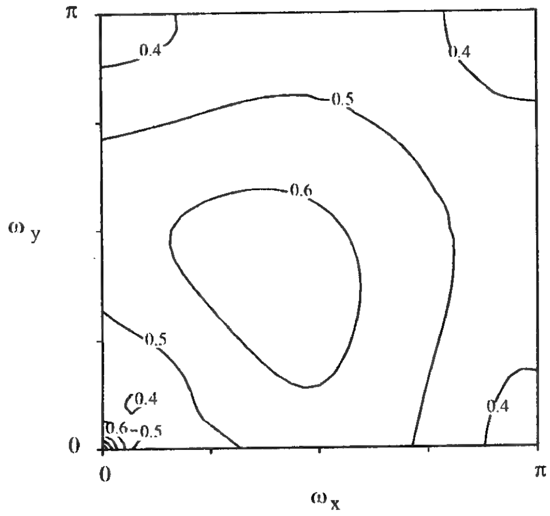


Figure 4: Euler: LGS-4, I/III, M=0.01, With Preconditioning, CFL=20

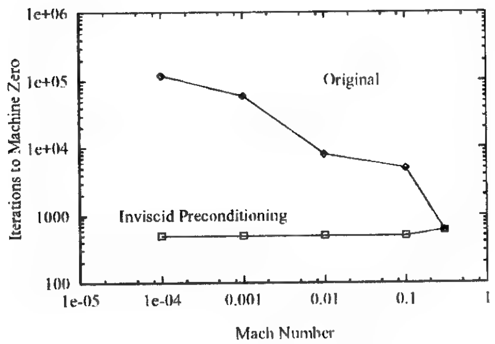


Figure 5: Effect of inviscid preconditioning on the number of iterations to converge to machine zero versus the Mach number.

independent of Mach number, and gives uniform convergence at all Mach numbers.

The significance of these stability results is easily shown by applying them to a simple flowfield consisting of inviscid flow in a straight duct. Figure 5 shows the convergence of the ADI system from an initial condition corresponding to a small perturbation from the exact (uniform flow) solution. Although this problem appears trivial, the return to uniform flow at low Mach numbers takes thousands of iterations without preconditioning, but is independent of Mach number with preconditioning. The number of iterations required for convergence without preconditioning is inversely related to the square of Mach number, and at $M = 10^{-5}$, some 10^5 iterations are required to reach convergence to machine error. When preconditioning is used, the number of iterations required for convergence is independent of Mach number and is similar to the number required for the non-preconditioned case at transonic conditions. The actual convergence rates for some of these cases are shown in Fig. 6. Similar preconditioned and non-preconditioned results are observed for the line Gauss-Seidel, upwind system. Applications to a wide range of practical problems have been demonstrated elsewhere [11-14], giving ample evidence that the Euler equation problem is well in hand.

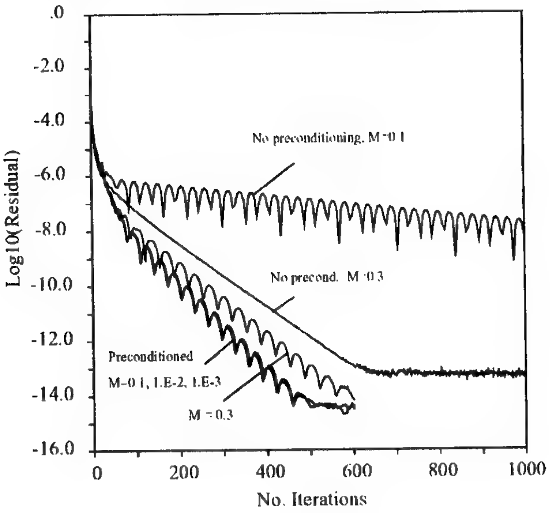


Figure 6: Convergence of the inviscid straight duct case at various Mach numbers using the original equations and the preconditioned equations.

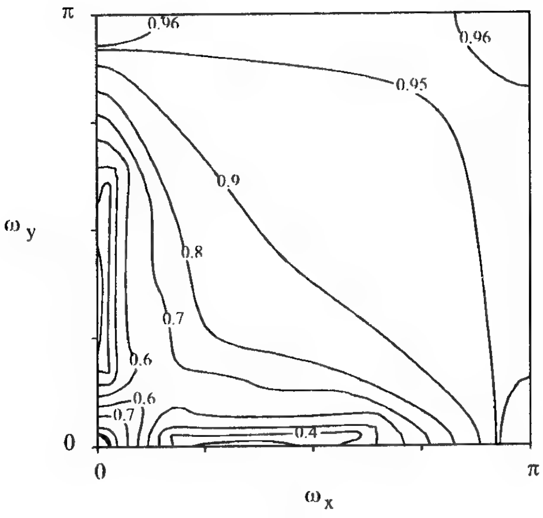


Figure 7: Navier-Stokes Eqns.: CD-ADI, M=0.001, $Re_{\Delta x}$=0.1, Viscous Preconditioning, CFL=5, VNN=5

To complete the stability survey for the Euler equations, we note that the stability results for the artificial compressibility version of the incompressible equations are identical to the low Mach number results in Figs. 2 and 4. These results clearly demonstrate the ability of the preconditioning method to apply equally well to compressible and incompressible solutions.

Stability results for the preconditioned Navier-Stokes equations are given on Fig. 7. These results are for the central-differenced ADI system at a cell Reynolds number of 0.1 and a Mach number of 0.01. (Note results for high cell Reynolds numbers are identical to the Euler results given in Fig. 2.) The viscous preconditioning is not quite as effective in controlling the stability eigenvalues as in the case of the Euler equations, but it still improves the stability map dramatically as compared to non-preconditioned results. Eigenvalues over most of the domain are around 0.9 with some increase toward the higher wavenumbers that arises because of the absence of diffusion in the continuity equation. The addition of artificial diffusion in continuity eliminates this difficulty and provides good viscous convergence as is shown next. Comparison with stability results based on the non-preconditioned equations or preconditioning with ρ'_p set to its inviscid value rather than its viscous value shows a substantial deterioration in eigenvalues for either case. Without preconditioning, the eigenvalues become very stiff, and while inviscid preconditioning changes the stability eigenvalues, it doesn't improve them. Clearly, viscous preconditioning is needed as the cell Reynolds number decreases.

Figure 8 demonstrates the effectiveness of the viscous preconditioning for the Navier Stokes equations for a second simple problem, that of fully developed flow in a pipe. Again, the initial condition corresponds to the exact solution plus a small perturbation. The figure shows the number of iterations required to converge to machine accuracy for cell Reynolds numbers ranging from 10^{-5} to 10. With viscous preconditioning, convergence is seen to be independent of cell Reynolds number over the entire spectrum. Solutions with inviscid preconditioning show a dramatic slowdown in convergence at the smaller Reynolds numbers, while computations with no preconditioning were

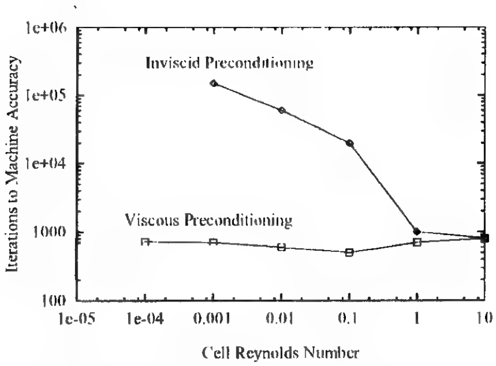


Figure 8: Convergence of the viscous straight duct case at various Cell Reynolds numbers using inviscid preconditioning or viscous preconditioning.

very irregular, and frequently did not converge. The flow Mach number for these calculations is taken as 10^{-3}.

7 REPRESENTATIVE SOLUTIONS FOR PRACTICAL PROBLEMS

Thus far we have shown how to transform the time derivatives so that the equations are well conditioned over the entire Reynolds-number / Mach-number regime. We have then used stability results for both the Euler and the Navier-Stokes equations to verify that these preconditioned equations provide effective damping factors. In addition, we have also shown that the preconditioning equations provide uniform convergence for simple problems at all Reynolds numbers and Mach numbers. The ultimate proof of convergence enhancement must, however, rest upon demonstration of effectiveness in practical problems. Over the past several years we have applied these systems to a broad variety of applications including low speed compressible flows, combustion problems, incompressible flows, supercritical fluids and extrusion modeling. Space does not permit a complete demonstration of all these examples, but we present some representative results to demonstrate the capabilities.

Figure 9 shows results for laminar flow over a backstep at a Reynolds number of 200. The *u*-velocity contours are shown, along with the convergence rate with the preconditioned and non-preconditioned cases. These computations are done with the line Gauss-Seidel algorithm. Clearly, viscous preconditioning provides a major enhancement to the convergence rate.

As a second example, we consider the flow through a converging diverging, rocket nozzle. The turbulent boundary layers in this nozzle are very thin because of the high Reynolds number, and strong wall cooling. The corresponding strong grid stretching (aspect ratios larger than 10^6) required near the wall introduces important low Reynolds number effects in this otherwise high Reynolds number flow. With standard algorithms, the solution converges at a reasonable rate for about four orders of magnitude (which would appear to be sufficient), and switches to a very slow rate of convergence. With preconditioning, the convergence continues to machine zero at a rate that is faster than the initial convergence of the non-preconditioned solution. The heat flux to the wall is shown in Fig. 10 as a function of axial distance for both calculations at several time steps. The lower plot shows that the preconditioned solution gives reasonable results after only 200 iterations, while after 400 iterations, the heat flux is indistinguishable from the machine-accuracy results. The standard algorithm produces very different results. After 2000 iterations (which corresponds to three orders of magnitude reduction in the global residuals), the wall heat flux is only about half its final converged value, and it takes more than 20,000 time steps to come within plotting accuracy of the fully converged result. When converged to machine accuracy, both the standard and the preconditioned algorithms give identical results. This shows that low Reynolds number cells near the wall (which determine the wall heat flux) can also totally control the overall convergence of the solution.

As a final example, we present a simulation of the flow in a uni-element gaseous rocket. The flowfield is generated by two co-annular jets entering through the left end of a cylinder whose diameter is 50 mm. The gas in the inner stream is oxygen, while that in the outer stream is hydrogen. The outer diameter of the hydrogen jet is 12 mm, giving a 38 mm backstep past which the jets exapand. The diameter of the oxygen jet is 8.4 mm, the hydrogen annulus is 1 mm, and the two are separated by a sleeve of thicknces 0.8 mm. An overall picture of the flowfield is given in Fig. 11. The back step generates a large recirculating region near the outer wall. The two gaseous streams begin to mix upon exiting the injector, but the finite thickness of the sleeve generates a small wake in which the hydrogen and oxygen first start to mix. This mixing region is very important to the computation because it acts as the primary flame-holding mechanism for the resulting diffusion flame (although the present results are for non-reacting flow).

Initial attempts at computing this flowfield with preconditioning showed very poor convergence. Although

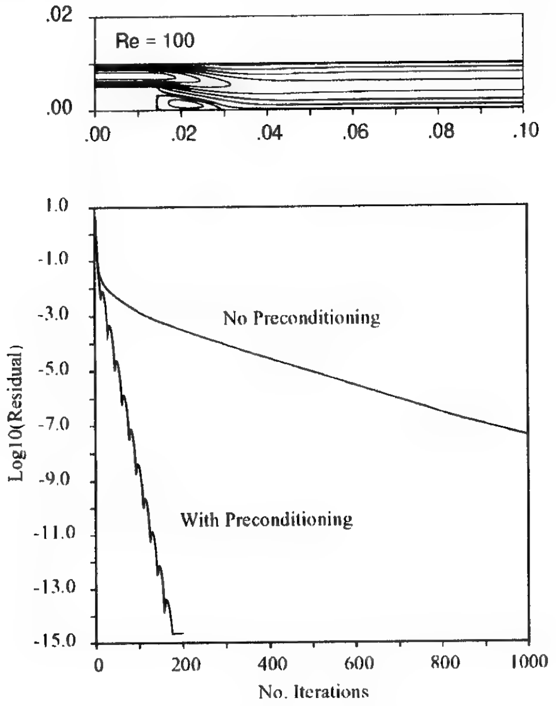
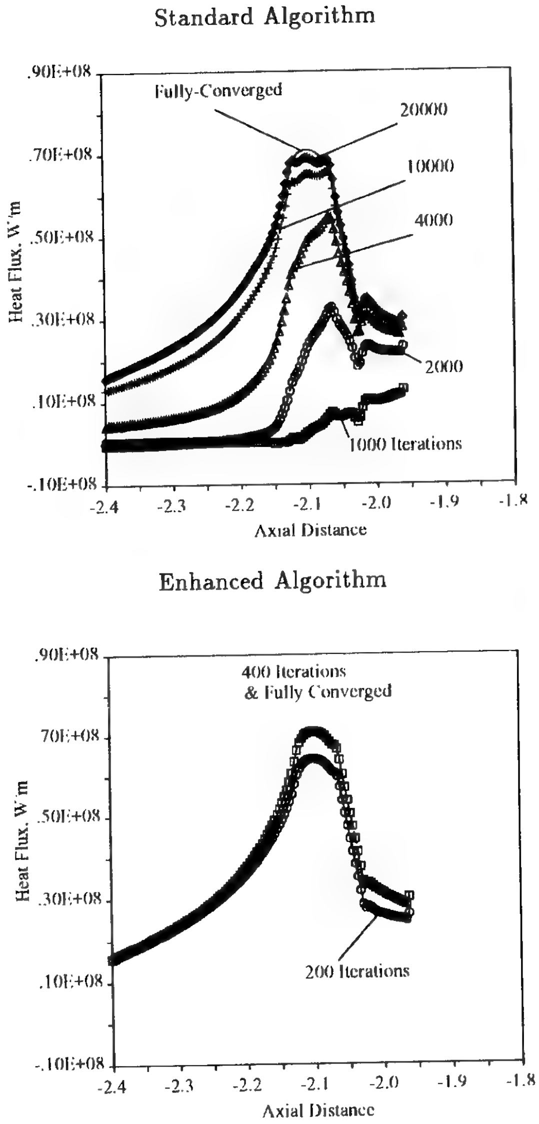


Figure 9: Contours of velocity and convergence for the backward-facing step at a Reynolds number of 100 using the four-sweep Line Gauss-Seidel scheme.



Standard Algorithm

Enhanced Algorithm

Figure 10: Temporal convergence of the heat flux along the nozzle wall for both the standard algorithm and the enhanced algorithm.

the non-preconditioned system did not converge well, it did converge slightly better than the preconditioned one. Careful investigation revealed that the reason for the convergence difficulty was because a steady solution failed to exist. The resulting flowfield was oscillatory in nature as determined by experimental observations. In addition, the computations showed the unsteadiness increased in strength as the grid was refined. The reason the preconditioned system showed poorer convergence was that it introduced a smaller amount of artificial dissipation than did the non-preconditioned system. (All computations were run with upwind flux difference splitting.) The velocity contours in the resulting unsteady solution is presented in Fig. 11 at three different instants of time.

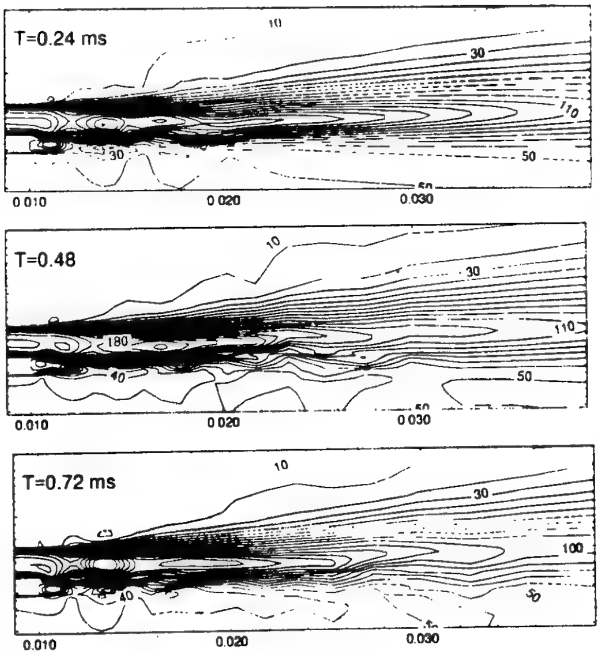


Figure 11: Unsteady Velocity Field near Injector Post for Hydrogen/Oxygen at O/F = 4.

The source of the unsteadiness appears to originate in the recirculating zone in the wake of the finite thickness sleeve between the two inlet streams. A close-up view of this region is given in Fig. 12. The details show that in this recirculating region, the heavier oxygen from the lower stream makes up the primary content of the recirculating region. The hydrogen mixes with the oxygen only along the upper side of the recirculating region. When the recirculation region sheds a vortex, it induces a substantial unsteadiness in the lighter hydrogen stream, which is then propagated into the recirculation region downstream of the backstep so that the entire flowfield oscillates in response to this narrow wake region.

Plots of the time rate of change of the velocity at a particular point near the wake region are given on Fig. 13. Even in the unsteady solution, the preconditioned results show larger amplitudes than do the non-preconditioned solutions. This is again because of a diminished amount of artificial damping. The impacts of increased artificial dissipation are shown by the first-order upwind results which are nearly steady. Comparisons with analytical solutions for simple shear layers indicate that the preconditioned results are more accurate.

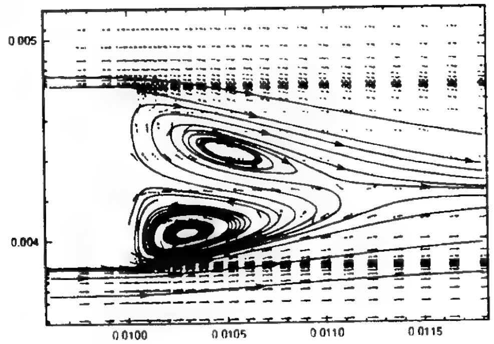


Figure 12: Velocity Vector/Streamline Field near Injector Post for Hydrogen/Oxygen Calculation

One issue with regard to preconditioned systems is their impact on code robustness. We have encountered many experiences in which preconditioning improves robustness (i.e., cases where the non-preconditioned code fails to converge, while preconditioning makes convergence very reliable). It must, however, be recognized that preconditioning increases the local time step dramatically and this large time step may require some restriction at early stages of the computation (although the restricted time step may still be larger than the corresponding non-preconditioned time step). The present example, however, demonstrates that there are some cases for which the preconditioning may not improve convergence because a steady solution does not exist. In these cases, the preconditioned system proves its worth in an unsteady, iterative solution procedure.

8 CONCLUSIONS

The proper limiting forms of the equations of motion at low speeds and in diffusion-dominated regions have been obtained by perturbation expansions and used as the basis for defining a preconditioning matrix for convergence enhancement. The expansion results show that ρ_p is the most important variable in controlling convergence while ρ_T and h_p are of secondary importance. Convergence control can be obtained by replacing these physical derivatives by artificial ones in the time derivatives, while retaining the physical quantities in the flux terms so the solutions are unchanged. Appropriate replacement terms for these quantities obtained from the expansion procedures are then generalized so that they approach the physical quantities in the transonic and supersonic regimes.

Following the development of a generalized preconditioner that ensures that the condition number of the Jacobian matrices of the equations of motion remain of order one at all Mach numbers, the resulting convergence characteristics are first checked by means of stability theory. The effectiveness of the methods is then verified by computations of a variety of problems, starting first with simple applications and then going to practical examples. Efficient, uniform convergence is demonstrated for a variety of applications covering a range of Reynolds and Mach number conditions. Overall, it is demonstrated that convergence enhancement of the Euler equations at low speeds is quite easy and can be readily ensured. Extension to the Navier-Stokes equations requires more care, but the present procedure provides much improved convergence rates in some of the traditional problem areas for time-marching methods, while having no detrimental effects in regimes where the methods already work efficiently.

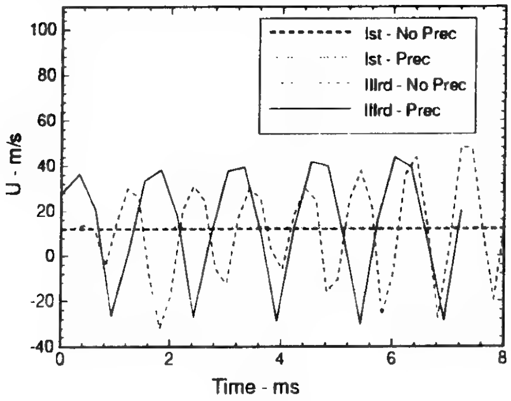


Figure 13: Time History of Axial Velocity at one point in the Injector Flowfield for various differencing schemes

ACKNOWLEDGMENT

This work was supported under NASA grants (NAGW-1356), (NAS 8-3886), and (NCC 8-46).

REFERENCES

[1] Patankar, S.V. (1980) Numerical Heat Transfer and Fluid Flow Series in Computational Methods in Mechanics and Thermal Sciences, McGraw-Hill Book Company.

[2] Chorin, A.J. (1967) "A Numerical Method for Solving Incompressible Viscous Flow Problems, Journal of Computational Physics, 2: 12-26.

[3] Rogers, S.E., Kwak, D. and Kiris, C. (1989) Numerical Solution of the Incompressible Navier-Stokes Equations for Steady-State and Time-Dependent Problems, AIAA Paper 89-0463.

[4] Rehm, R.G. and Baum, H.R. (1978) J. Research of the National Bureau of Standards, 83, 297.

[5] Guerra, J. and Gustafsson, B. (1986) A Numerical Method for Incompressible and Compressible Flow Problems with Smooth Solutions, Journal of Computational Physics, 63, pp. 377-396.

[6] Merkle, C.L. and Choi, Y.H. (1985) Computation of Low-Speed Compressible Flows with Time-Marching Procedures, International Journal for Numerical Methods in Engineering, 25: 293-311.

[7] Withington, J.P., Shuen, J.S. and Yang, V. (1991) A Time Accurate, Implicit Method for Chemically Reacting Flows at All Mach Numbers, AIAA Paper 91-0581.

[8] Fiterman, A., Turkel, E. and Vatsa, V. (1995) Pressure Updating Methods for the Steady-State Fluid Equations, AIAA Paper 95-1652.

[9] Viviand, H. (1985) Pseudo-Unsteady Systems for Steady Inviscid Calculations, Numerical Methods for the Euler Equations of Fluid Dynamics SIAM, pp. 334-368.

[10] Turkel, E. (1987) Preconditioned Methods for Solving the Incompressible and Low Speed Compressible Equations, Journal of Computational Physics, 72, 277-298.

[11] Venkateswaran, S., Weiss, J.M., Merkle, C.L. and Choi, Y.H. (1992) Propulsion-Related Flowfields Using the Preconditioned Navier-Stokes Equations, AIAA Paper 92-3437.

[12] Choi, Y.H. and Merkle, C.L. (1993) The Application of Preconditioning to Viscous Flows, Journal of Computational Physics, 105: 207-223.

[13] Shuen, J.S., Chen, K.H. and Choi, Y.H. (1992) A Time-Accurate Algorithm for Chemical Non-Equilibrium Viscous Flows at All Speeds, AIAA Paper 92-3639.

[14] Weiss, J.M. and Smith, W.A. (1994) Preconditioning Applied to Variable and Constant Density Flows on Unstructured Meshes, AIAA Paper 94-2209.

[15] van Leer, B., Lee, W.T. and Roe, P.L. (1991) Characteristic Time-Stepping or Local Preconditioning of the Euler Equations, AIAA Paper 92-1552-CP.

[16] Godfrey, A.G., Walters, R.W. and van Leer, B. (1993) Preconditioning for the Navier-Stokes Equations, AIAA Paper 93-0535.

Practical Aspects of Krylov subspace Iterative Methods In CFD

Thomas H. Pulliam[*]
Advanced Computational Methods Branch

Stuart Rogers [†]
Design Cycle Technologies Branch

Timothy Barth [‡]
Advanced Computational Methods Branch

NASA Ames Research Center Moffett Field, CA 94035-1000, USA.

September 13, 1995

Abstract

Implementation issues associated with the application of Krylov subspace iterative methods, such as Newton-GMRES, are presented within the framework of practical CFD applications. This paper will categorize, evaluate and contrast the major ingredients (function evaluations, matrix-vector products and preconditioners) of Newton-GMRES Krylov subspace methods in terms of their effect on the local linear and global nonlinear convergence, memory requirements, and accuracy, The discussion will focus on Newton-GMRES in both a structured multi-zone incompressible Navier-Stokes solver and an unstructured mesh finite-volume Navier-Stokes solver. Approximate vs. exact matrix-vector products, effective preconditioners and other pertinent issues will be addressed.

1 Introduction

Interest in iterative methods in CFD has been motivated not only by the requirement for better convergence and speed of numerical codes, but also by the availability of faster, larger memory serial and parallel machines. The coupling of Newton's method with iterative solvers is an effective approach for solving the large systems of nonlinear equations which arise from discretized forms of the Euler and Navier-Stokes equations. One of the main motivations for the use of Newton's method is the possibility of superlinear (and in some cases quadratic) asymptotic convergence. References [7, 21] are examples of successful implementations of exact Newton's method for two-dimensional Navier-Stokes codes. Most of the conventional implicit schemes used today are effectively approximate-Newton methods. The approximations appear in the form of simplifications in the functional Jacobian or some form of under/over relaxation strategy, see e.g. [11] or [15]. In practice these simplifications are employed for reasons such as efficiency, implementation ease, or non-analyticity of operators (e.g., discrete limiters in differencing schemes based on Riemann solvers). Over a wide range of numerical methods developed for the solution of the multidimensional Navier-Stokes equations, the rigorous application of Newton's method would require the inversion of a large block banded matrix, which even by today's standards, poses many obstacles in terms of memory requirements and speed. An alternative to direct matrix inversion is the use of iterative matrix solution methods. In particular, the class of Krylov subspace methods known as GMRES [19] will be considered. Wigton [23] was the first to successfully implement GMRES for a two-dimensional Navier-Stokes code.

The difficulties associated with iterative methods such as GMRES lie in the rapid expansion of memory requirements inherent in the embedded Arnoldi process (storing the Krylov subspace vectors), the need to perform the matrix vector "Ap" products (which sometimes requires the storage of the matrix A), and the preconditioning of the system of equations by some approximate inverse of A to improve the convergence of GMRES.

The purpose of this paper is to focus on the implementation details and specific results from the application of Newton-GMRES to both a struc-

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

tured multi-zone incompressible Navier-Stokes code (INS2D, Rogers [17]) and an unstructured mesh Navier-Stokes code due to Barth [5]. We use these two codes as case studies, but the lessons learned, approximations assumed, results obtained, and general conclusion, are applicable to most implementations of Newton-GMRES to systems of PDE's.

2 General Formulation

In a general form, we cast the discrete approximation to the steady multi-dimensional Navier-Stokes equations as

$$\begin{aligned}\mathcal{R}(\mathbf{Q}) &= \mathcal{R}(\cdots, Q_{j,k-1}, Q_{j-1,k},\\ &\quad Q_{j,k}, Q_{j+1,k}, Q_{j,k-1}, \cdots) = 0, \qquad (1)\end{aligned}$$

where we are representing the support of the operator as involving neighboring points in a computational mesh and **Q** is the solution variable (typically the conserved variables). Although this representation appears in a structured mesh form and is rather compact (involving only three points in each computational direction), we intend it to also represent an unstructured mesh template and possibly higher order - higher dimensional - more broadband support. We shall refer to this as the **Function Evaluation** step in the overall process. For fixed point solutions (steady-state) we require the solution of $\mathcal{R}(\mathbf{Q}) = 0$.

A time accurate approach to the solution assumes the form of

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathcal{R}(\mathbf{Q}) = 0 \qquad (2)$$

which can either represent the artificial-compressibility scheme for the incompressible Navier-Stokes equations [17] or the full Navier-Stokes unstructured mesh scheme [5]. Applying implicit Euler time differencing with the usual Taylor series linearization in time we have

$$\left[\frac{D}{\Delta t} - \frac{\partial \mathcal{R}(\mathbf{Q})^n}{\partial \mathbf{Q}}\right] \Delta \mathbf{Q} = \mathcal{R}(\mathbf{Q})^n \qquad (3)$$

with $\Delta \mathbf{Q} = \mathbf{Q}^{n+1} - \mathbf{Q}^n$, $\frac{\partial \mathcal{R}(\mathbf{Q})^n}{\partial \mathbf{Q}}$ the Jacobian ($\mathcal{A}$) of the **Function Evaluation** , $\mathcal{R}(\mathbf{Q})$, and D a positive diagonal matrix. For $\Delta t \rightarrow \infty$ this is exactly Newton's method and for finite Δt a relaxed form of Newton's method. In many applications of Newton-like methods to the Euler and Navier-Stokes equations this time-like relaxation is used to start the solution process. A finite time step Δt is used initally to get past the rather violent nonlinear startup and then increased to $\Delta t \rightarrow \infty$ leading to rapid linear, super-linear or quadratic convergence depending on the characteristics of the Newton solver, e.g.[21] or [5]. It is convenient to recast Eq. 3 in the general form

$$\mathbf{b} - \mathcal{A}\mathbf{x} = 0, \qquad (4)$$

where $\mathbf{b} = \mathcal{R}(\mathbf{Q})$, $\mathbf{x} = \Delta \mathbf{Q}$ and $\mathcal{A}$ is a matrix operator.

The numerical process involved in solving Eq˙ 3 will be referred to here as the **Inner Iteration** at a particular step n. The overall iteration of the nonlinear system will be referred to as the **Outer Iteration** . There are a number of successful approaches to the **Inner Iteration**. In the case of structured mesh applications, $\mathcal{A}$ represents a sparse block banded matrix which can be solved with various methods such as point or line relaxation [17] or approximate factorization, e.g.[14]. In the unstructured mesh case, $\mathcal{A}$ may not have a simple underlying structure, but the **Inner Iteration** can be successfully solved with a wide variety of relaxation techniques [22], [5]. For the present discussion we shall focus on the GMRES Krylov projection technique for solving the **Inner Iteration**.

The GMRES (Generalized Minimal RESidual) method was introduced by Saad and Schultz [19] for solving large sparse systems of linear equations. The GMRES algorithm is a Krylov subspace method where given a matrix $\mathcal{A} \in \Re^{N\times N}$, a vector $v \in \Re^N$ and an integer $m \geq 1$, the Krylov subspace associated with $\mathcal{A}$, v and m is defined as

$$K_m(\mathcal{A}, v) = span\{v, \mathcal{A}v, \mathcal{A}^2 v, \cdots, \mathcal{A}^{m-1}v\}. \qquad (5)$$

In the GMRES algorithm an initial guess x_0 to the solution of the linear system is given from which the initial residual is defined

$$r_0 = \mathbf{b} - \mathcal{A}\mathbf{x}_0. \qquad (6)$$

The GMRES method then attempts to find $z_m \in K_m(\mathcal{A}, r_0)$ such that the residual vector $\mathbf{b} - \mathcal{A}(\mathbf{x}_0 + z_m)$ is small . This is done so that at each iteration the residual norm is minimized. One important parameter for the GMRES method is the size of the subspace m. As m increases, the memory increases linearly and the computation quadratically. The parameter m is usually chosen based on storage requirements and effectiveness of the **Inner Iteration**. In the discussion below, we will have more specific things to say about this requirement and it's effect on the overall process. To avoid the increasing memory and computation requirements with increasing m, a common modification of GMRES is to apply restarts. An upper bound m_r on m is chosen and if convergence is not reached, the Krylov subspace process is restarted with the current residual r_{m_r} replacing r_0. In this case, the memory requirements are traded off against the convergence of the **Inner Iteration** process, and this will definitely affect the **Outer Iteration** .

Numerical experience has shown that success or failure of GMRES hinges critically on adequate preconditioning of the linear systems to be solved. A preconditioning matrix M is usually applied in either a left-preconditioning $M(\mathbf{b} - \mathcal{A}\mathbf{x}) = 0$ or right-preconditioning $\mathbf{b} - \mathcal{A}M\mathbf{y} = 0$ (with $\mathbf{y} = M^{-1}\mathbf{x}$) fashion. Ideally M should be chosen to be an approximation to $\mathcal{A}^{-1}$. Although, the most successful and popular form of M appears to be ILU [13] (Incomplete Lower-Upper Factorization), we will also consider alternate preconditioners in the next section.

The important ingredients of the Newton-GMRES method which we will focus on in this paper are the Ap products required to form the Krylov subspace vectors $K_m(A, v)$, the choice of the preconditioner M, the size of the subspace m and restart size m_r, and the storage requirement influenced by all these factors. We will attempt to put the various trade offs in terms of memory requirements, convergence and efficiency in perspective, (in particular for the two approaches discussed here, but also in general).

3 Structured Mesh Incompressible Navier-Stokes

Rogers [17] has implemented the Newton-GMRES algorithm into a two-dimensional incompressible Navier-Stokes code (INS2D) and has made some significant comparisons with the conventional techniques of implicit point and implicit Gauss-Seidel line relaxation. The INS2D flow code [18] solves the Reynolds-averaged incompressible Navier-Stokes equations using the method of artificial compressibility, [9]. It is capable of handling multiple-zone structured grids using either a patched multi-block (pointwise continuous) interface, or an overlaid (chimera) interface between zones. The boundary conditions at the physical boundaries and at zonal boundaries are applied in an implicit fashion during the solution process. A third-order, upwind-differencing scheme based on the method of Roe [16] is used to descritized the convective terms, and the viscous terms are differenced using second-order central differences. The system of equations is integrated in pseudo-time using an implicit Euler time discretization. Typically, the time step is set to infinity (10^8) which results in a Newton's method approach where the implicit point or line relaxation schemes are used for the **Inner Iteration** or more specific to this paper, GMRES is used for the **Inner Iteration**.

In the current implementation, the Jacobian $\mathcal{A}$ is formed based on a first-order differencing of the convective terms, whereas third-order differencing is used for $\mathcal{R}(\mathbf{Q})$. In addition, approximate Jacobians of the Roe flux differences from the upwind-differencing scheme are used in the definition of $\mathcal{A}$, see [1] for more details. The first-order difference operator is used to reduce the bandwidth of the resulting $\mathcal{A}$ matrix, which has lower memory and computational requirements for the solution of Eq. 3. However, this use of approximate Jacobians can also slow the convergence to a steady state, that is, the **Outer Iteration** nonlinear Newton process is affected.

The GMRES implementation is preconditioned using block ILU(0) [13] and the matrix $\mathcal{A}$ is stored so that Ap products can be efficiently formed and the ILU process streamlined. For comparison, block point relaxation and block line relaxation are used as both the **Inner Iteration** solver and as preconditioners for the GMRES **Inner Iteration** process. Including a subspace size typically on the order of $m = 10$ leads to additional storage requirements as discussed below which are somewhat of a burden in two dimensions and would be a significant hindrance in three dimensions. The use of the approximate Jacobian (due to the first order form and the linearization errors associated with the Roe solver) produces an approximate Newton's method and therefore linear convergence is realized as opposed to the potential for quadratic convergence.

Rogers [17] examines a wide range of cases and options in his paper on the Newton-GMRES implementation. Table 1 shows the characteristics of the cases presented and itemizes the costs of the various schemes for each case broken down by the fundamental steps in the algorithm. Base memory (B MW) includes all overhead storage for the algorithm including memory for either L (line relaxation), P (point relaxation) or the Ap product in G (GMRES) ($\approx$ 76 words/point). The additional memory (A MW) is composed of subspace size ($\approx 3 \times (m + 4)$ words/point) and preconditioner ($\approx$ 9 words/point) contributions for GMRES. The timings are in ms/pt : milliseconds/point to convergence, (maximum divergence $\approx 10^{-8}$). The standard approaches of point relaxation and line relaxation are compared directly with the Newton-GMRES scheme and are also assessed as preconditioners for Newton-GMRES. The first few cases are for a NACA 4412 airfoil an angle of attack $\alpha = 13.87°$ and a Reynolds number, $Re = 1.5 \times 10^6$ and are computed on a set of refined grids. The multi-element case is a three element airfoil at $\alpha = 8°$ and $Re = 9 \times 10^6$, a schematic of the grid system is shown in Figure 1.

Figures 2, 3, 4, 5 show comparisons for the above cases, where the symbols represent every 50 **Outer Iteration** . It seems obvious from these results that the GMRES-ILU combination is the more efficient in terms of computation time. On the other hand, the negative aspect of the GMRES-ILU combination is the memory requirements. By examining the trade offs between CPU time to convergence and memory requirements optimal choices can be made.

For example, Figure 2 shows the effect on CPU time to convergence for various choices of m. A subspace size $m = 10$ seems to be optimal in terms of computational costs, including reasonable added memory requirements. Also, note that for the converging cases of $m = 10, 20, 40$, it required 50 **Outer Iteration** to reach the same level of convergence (CPU times are larger reflecting the added computational costs of a larger subspace size). This is not surprising since the inexact Jacobian used in this scheme limits the **Inner Iteration** process to linear convergence. Therefore, after some point, it does not pay to converge the **Inner Iteration** past some tolerance level without incurring additional cost in terms of CPU time and memory.

Case	Method	B MW	A MW	ms/pt
Airfoil	L(5)	0.28	0.006	1.98
Grid 1	P(20)	0.28	0.001	2.31
119x31	G(10)+LR	0.28	0.161	3.17
	G(10)+PR	0.28	0.156	2.12
	G(10)+ILU	0.28	0.188	1.14
Airfoil	L(10)	1.10	0.011	3.68
Grid 2	P(20)	1.10	0.002	5.13
237x61	G(5)+L	1.10	0.618	3.77
	G(10)+P	1.10	0.609	4.48
	G(10)+ILU	1.10	0.737	1.45
Airfoil	L(10)	4.35	0.023	8.79
Grid 3	P(40)	4.35	0.004	12.56
473x121	G(10)+ILU	4.35	2.920	3.91
Multi-	L(10)	5.17	0.015	49.7
Element	P(20)	5.17	0.003	14.7
68K pts	G(10)+ILU	5.17	3.468	5.37

Table 1: Cost comparisons of iterative methods for INS2D for various cases and schemes. L(n): Line Relaxation for n iterations, P(n): Point Relaxation for n iterations, G(m)+ X : GMRES with subspace size m using scheme X for preconditioner.

Figure 6 shows the effect of decreasing the **Inner Iteration** tolerance level ϵ thereby solving the GMRES step more accurately. In this case, to reach a certain convergence criteria, e.g.**Outer Iteration** residual to 10^{-8}, in the least number of iteration, requires decreasing ϵ, $\epsilon = 10^{-5}$ gets there in 50 **Outer Iteration** . On the other hand, the CPU time cost (shown in milliseconds/point) and average subspace size m (which leads to addition memory requirements proportional to m) indicate that a loose tolerance, say $\epsilon \approx 10^{-1}$ and small m produce the most efficient combination. This leads to $m = 10$ as the optimal choice both in terms of CPU efficiency and memory requirements.

Memory estimates for the three-dimensional code INS3D include a base memory of 146 words/point, additional GMRES memory: $4 \times (m+4)$ words/point and preconditioner memory of 16 words/point. Thus for GMRES(10)+ILU(0) the total memory is 218 words/point. Examples include a simple wing: 0.2 million points (Mpoints): 43.6 MW, a wing+slat+flap: 1.6 Mpoints: 349 MW, and a C17 Aircraft: 25 Mpoints: 5450 MW = 5.45 GW. These requirements are excessive in three-dimensions and need to be reduced if these codes are to be used in practice.

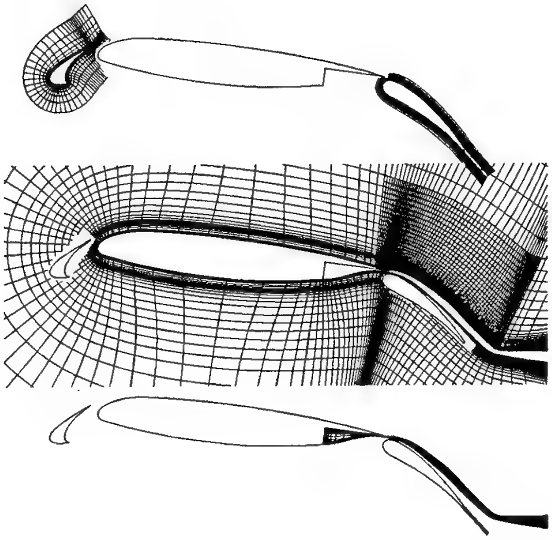


Figure 1: Grid around a three-element airfoil.

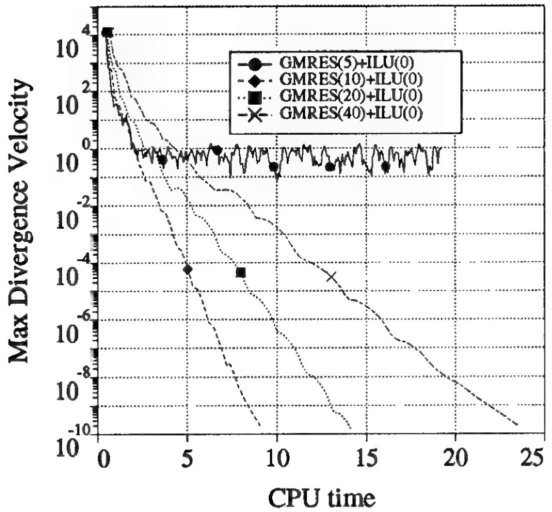


Figure 2: Results for Grid 1 showing effect of subspace size GMRES(m) on Newton-GMRES.

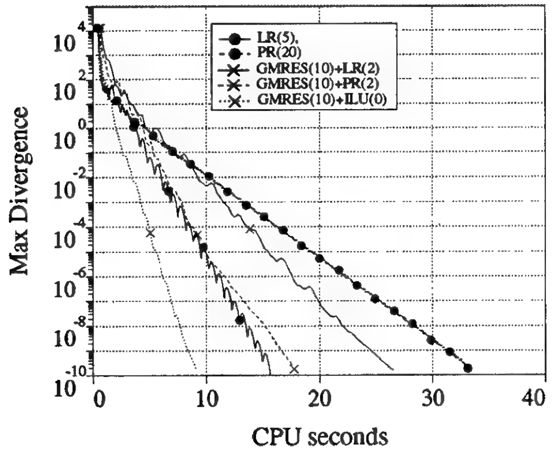


Figure 3: Results for Grid 1 Comparing Various Schemes.

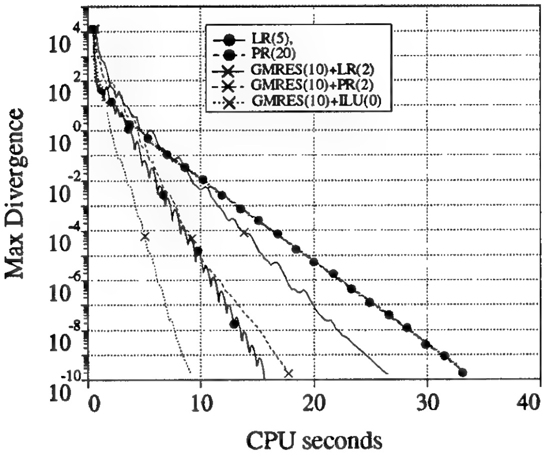


Figure 4: Results for Grid 2 Comparing Various Schemes.

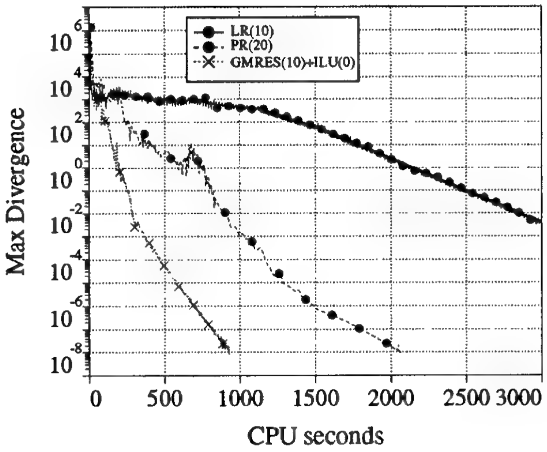


Figure 5: Results for Multi-Element Airfoil Comparing Various Schemes.

4 Unstructured Mesh Compressible Navier-Stokes

Barth [5] has implemented the Newton-GMRES algorithm into a two and three -dimensional unstructured mesh Navier-Stokes approach. In this case the flow equations are solved using an edge-based unstructured mesh quadrature scheme characterized as an approximate and/or exact Roe Riemann solver based on piecewise polynomial reconstruction, this defines the **Function Evaluation** . Details of the flow algorithm can be found in Barth [6, 2, 3]. The details of the Newton-GMRES implementation include exact Ap products for the second order discretization with a first or second order approximate $\mathcal{A}$ used only to construct the ILU preconditioner. Barth presents three methods to compute Ap products, one in which the exact Jacobian is stored (requiring a significant increase in memory requirements), a numerical evaluation using Fréchet derivatives [8] which is a matrix-free approach, and another matrix free approach using an exact product form where proper linearization of the Riemann solvers and the recon-

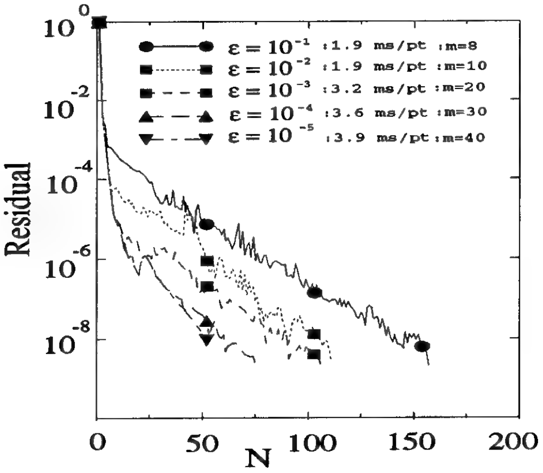


Figure 6: Comparison for various **Inner Iteration** tolerance levels for Grid 2 case. Tolerance levels are given along with milliseconds per point to convergence and average subspace size used m.

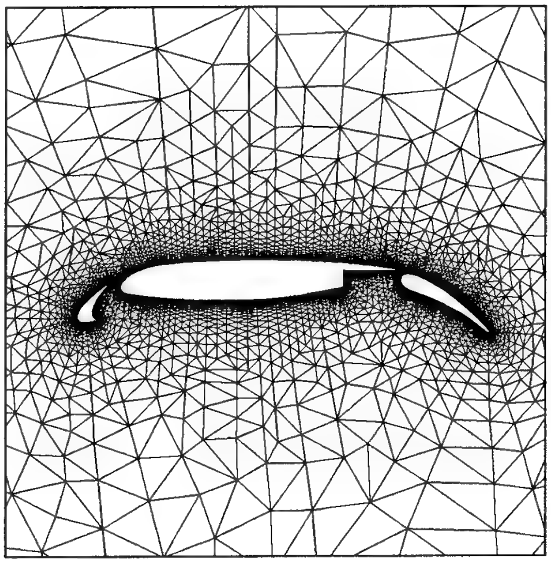


Figure 7: Multi-element airfoil triangulation, 22,000 vertices.

struction/quadrature mechanism of the residual vector assembly are used to produce the Ap product. Since exact Ap products are used, quadratic convergence can be realized. The preconditioned **Inner Iteration** is also fairly efficient, employing a subspace size on the order of $m_r = 12$ and a modest number of restarts. The resulting scheme can be mapped very successfully onto a parallel processor environment.

Figures 7,8,9,10 show an example computation for viscous flow with turbulence about the multiple-element airfoil geometry. This geometry has been triangulated using the Steiner triangulation algorithm described in [4], see Figure 7. The mesh contains approximately 22,000 vertices with cells near the airfoil surface attaining aspect ratios greater than 1000:1. This example provides a demanding test case for CFD algorithms. The experimental flow conditions are $M_\infty = .20$, $\alpha = 16°$, and a Reynolds number of 9^{-6}. Experimental results are given in [20] and computed results are shown in Figure 8. Even though the wake passing over the main element is not well resolved, the surface pressure coefficient shown in Figure 9 agrees quite well with experiment.

The convergence history shown in Figure 10 is typical for aerodynamic high lift computations.

Some of the more practical aspects from Barth's [5] implementation of Newton-GMRES are discussed below.

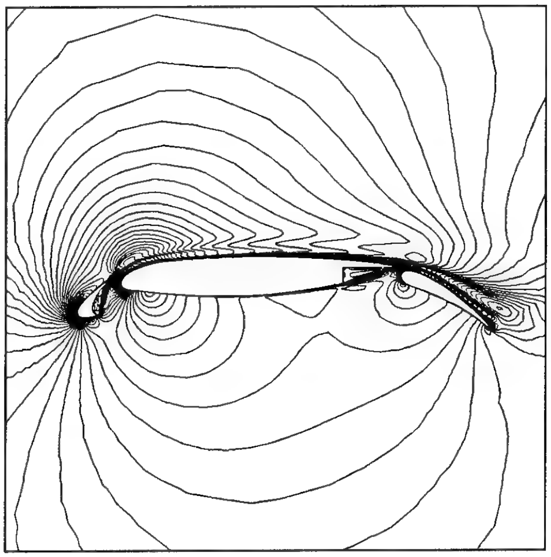


Figure 8: Multi-element airfoil solution isomach contours, $M_\infty = 0.2$, $\alpha = 16.0°$, $Re = 9.0$ million.

4.1 Storage Requirements

In practice we will be solving systems of l coupled equations so that each nonzero entry of the matrix is actually a small $l \times l$ block. The schemes employed require data from distance-one neighbors in

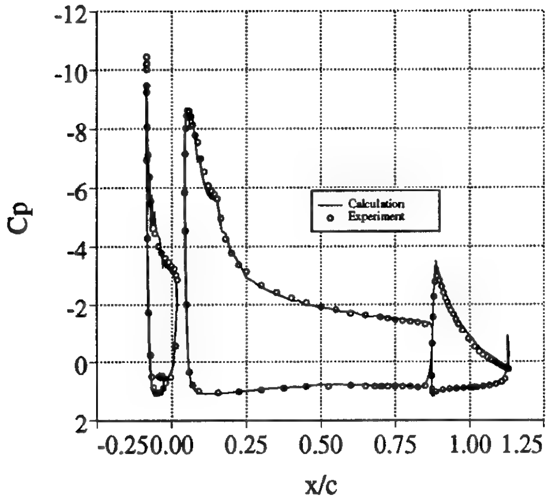


Figure 9: Comparison of computational and experimental surface pressure coefficients.

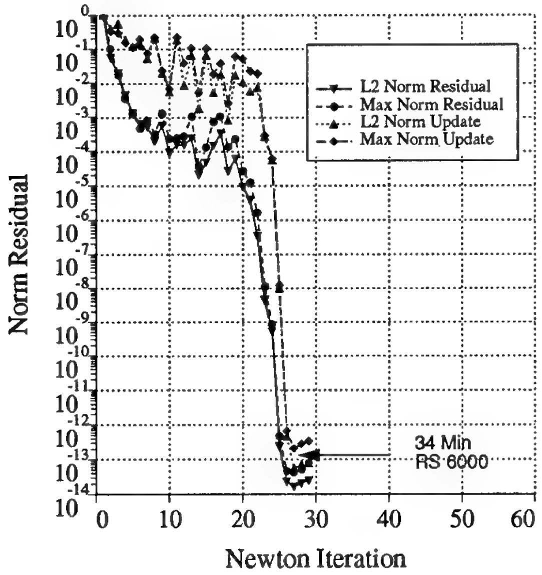


Figure 10: Solution convergence history.

the graph (mesh). In addition, the higher order accurate schemes require distance-two neighbors in building the scheme, see Barth [5, 3, 6]. First consider the situation in which the scheme requires only distance-one neighbors. The number of nonzero entries in each row of the matrix is related to the number of edges incident to the vertex associated with that row. Equivalently, each edge $e(v_i, v_j)$ will guarantee nonzero entries in the i-th column and j-th row and similarly the j-th column and i-th row. In addition, nonzero entries will be placed on the diagonal of the matrix. From this counting argument we see that the number of nonzero block entries, nnz, in the matrix is exactly twice the number of edges plus the number of vertices, $2E+N$ (approximately $7N$ in 2D). Table 2 (based on a similar counting argument) shows approximate requirements for storing distance-one and distance-two neighboring information as a sparse matrix.

Note that the entries of the sparse matrix associated with Newton's method (for solution of the Navier-Stokes equations and an associated 1 equation turbulence model) are actually small 5×5 and 6×6 blocks in two and three dimensions respectively. At first glance, this storage requirement appears prohibitively large. While this may be true to some extent today, the memory capacity of computers is expanding at a rapid rate. It is quite reasonable to expect that in the foreseeable future sufficient memory will be available for solving most problems of engineering interest. Even so, it is possible to reduce, and in some cases eliminate, the explicit storage of the Jacobian matrix without compromising the favorable convergence characteristics of Newton's method.

Dim.	nnz (Distance-1)	nnz (Distance-2)
2	$7N$	$19N$
3	$14N$	$55N$

Table 2: Storage Estimates for Sparse Matrices.

4.2 Calculating Analytic Jacobian Derivatives

In this section we address the task of computing Jacobian derivatives for Newton's method. In the following section we consider the related task of multiplying an arbitrary vector by the Jacobian matrix.

A major task in the overall calculation of the Jacobian derivatives for the finite-volume discretization is the linearization of the numerical flux vector with respect to the two solution states, e.g. given the Roe flux function [16]

$$\mathbf{h}(\mathbf{u}^R, \mathbf{u}^L; \mathbf{n}) = \frac{1}{2}\left(\mathbf{f}(\mathbf{u}^R, \mathbf{n}) + \mathbf{f}(\mathbf{u}^L, \mathbf{n})\right) \quad (7)$$

$$- \frac{1}{2}|A(\mathbf{u}^R, \mathbf{u}^L; \mathbf{n})|\left(\mathbf{u}^R - \mathbf{u}^L\right) \quad (8)$$

we require the Jacobian terms $\frac{d\mathbf{h}}{d\mathbf{u}^R}$ and $\frac{d\mathbf{h}}{d\mathbf{u}^L}$. Here, $\mathbf{f}$ is the flux function, $\mathbf{n}$ a geometric normal, $A = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}$, the physical flux Jacobian, evaluated at some combination of the right and left states of the flow variables, $\mathbf{u}^R, \mathbf{u}^L$. Exact analytical expressions for these terms are available [1]. In constructing the Jacobian matrix for the entire scheme it is useful to conceptualize the finite-volume scheme in *composition form*:

$$\mathbf{R}(\mathbf{Q}) = \mathcal{L}_1(\mathcal{L}_2(\mathbf{Q})), \tag{9}$$

with $\mathcal{L}_1$ representing the flux quadrature and accumulation step and $\mathcal{L}_2$ representing the data reconstruction step. In this form, each operator requires distance-1 information. The Jacobian matrix can then be written as

$$\frac{d\mathbf{R}}{d\mathbf{Q}} = \frac{d\mathcal{L}_1}{d\mathcal{L}_2}\frac{d\mathcal{L}_2}{d\mathbf{Q}} \tag{10}$$

with the critical observation that the Jacobian matrix can be calculated as the sparse product of two matrices. This could potentially be an expensive task, but because of the special form of $\mathcal{L}_1$ and $\mathcal{L}_2$, the resulting sparse product produces at most distance-2 fill and can be computed at reasonable cost.

4.3 Exact and Approximate Jacobian Matrix-Vector Products

Consider the standard matrix equation $\mathbf{b} - \mathcal{A}\mathbf{x} = 0$. Iterative matrix solution algorithms for this problem requires the computation of matrix-vector products of the form $\mathcal{A}\mathbf{p}$ for some arbitrary $\mathbf{p}$ vector. In the approximate Newton algorithm

$$\mathcal{A} = \left[\frac{D}{\Delta t} - \frac{d\mathbf{R}}{d\mathbf{Q}}\right] \tag{11}$$

where D is a positive diagonal matrix. In practice the diagonal entries are locally scaled as a exponential function of the norm of the residual

$$\frac{D_i}{\Delta t} = \frac{\mathrm{cfl}_i}{\mathrm{cfl}_{max}}, \qquad \mathrm{cfl}_{max} = f(\|\mathbf{R}(\mathbf{Q})^n\|)$$

so that when $\|\mathbf{R}(\mathbf{Q})\| \to 0$, $\mathrm{cfl}_{max} \to \infty$ and the scheme approaches Newton's method. It should be emphasized that by using this strategy, the scheme is technically an approximate Newton method which becomes exact only in the final few iterations of the computation.

A major step in the matrix-vector product $\mathcal{A}\mathbf{p}$ is the computation of Jacobian derivatives in the direction of $\mathbf{p}$ (a Fréchet derivative)

$$\mathcal{A}\mathbf{p} = \frac{D}{\Delta t}\mathbf{p} - \frac{d\mathbf{R}}{d\mathbf{Q}}\mathbf{p}. \tag{12}$$

Several possible strategies exist for computing the needed Fréchet derivatives:

4.3.1 Sparse Matrix-Vector Multiply

The most straightforward strategy is to analytically compute and store the Jacobian matrix using a compressed storage scheme designed for sparse matrices. This strategy has the added benefit that a copy of the matrix can also be used as a preconditioner for the iterative solver. In addition, the explicit storage also permits the formation of the transposed matrix problem which is often encountered in optimization procedures coupled with Newton's method. Obviously, a drawback of this approach is the large storage requirement.

4.3.2 Approximate Fréchet Derivatives

An alternative to analytically calculating Fréchet derivatives is to approximate them using finite differences, [12] [8] [10]. The required Fréchet derivative is a limiting form of the difference approximation

$$\frac{d\mathbf{R}}{d\mathbf{Q}}\mathbf{p} = \lim_{\epsilon\to 0}\frac{\mathbf{R}(\mathbf{Q}+\epsilon\mathbf{p}) - \mathbf{R}(\mathbf{Q})}{\epsilon}.$$

The primary concern with this approach is the accuracy of derivatives and the optimal choice for ϵ. If derivatives are not computed accurately then methods such as GMRES iteration may stall or fail. Using a forward difference approximation, ϵ must be carefully chosen. In general it is insufficient to choose ϵ as a constant such as the square root of machine precision. Johan [12] also mentions this fact and gives some analysis for choosing ϵ but this analysis assumes that $\mathbf{R}(\mathbf{Q})$ is well scaled. A common choice for ϵ is given by

$$\epsilon = \delta_0 + \delta_1\frac{\|\mathbf{Q}\|}{\|\mathbf{p}\|} \tag{13}$$

with suitably chosen constants δ_0 and δ_1. An alternative to forward differencing is to use higher order accurate formula such as central differencing at double the computational cost.

The clear attraction of this approach is the low memory requirement. On the other hand, the numerical computation of Fréchet derivatives does not produce a matrix approximation which can be used to precondition the system.

4.3.3 Exact Product Forms

In this section we will present a technique for constructing matrix-vector products which is an exact calculation of the Fréchet derivative. Extension to systems and the inclusion of diffusion terms are also handled using this technique.

Let $G(E, V)$ denote the triangulation in 2D or 3D with n vertices and m edges. Next we define the incidence matrix

$$\mathcal{C}_{il} = \begin{cases} -1 & \text{if } v_i \text{ is the origin of edge } l \\ 1 & \text{if } v_i \text{ is the destination of edge } l \\ 0 & \text{otherwise} \end{cases} . \tag{14}$$

Let $\mathbf{h} = \mathbf{h}(\mathbf{u}^L, \mathbf{u}^R; \mathbf{n})$ denote the numerical flux function as defined by Equation 8. For a system of l coupled differential equations, the Jacobian matrix entries are actually small $l \times l$ blocks. For ease of exposition, we tacitly treat these small blocks as scalar entries. Under these simplifications, the desired matrix-vector product is given by

$$\frac{d\mathbf{R}}{d\mathbf{Q}}\mathbf{p} = \mathcal{C}^T \left[\left[\frac{d\mathbf{h}}{d\mathbf{u}^L}\right] \left[\frac{d\mathbf{u}^L}{d\mathbf{u}}\right] + \left[\frac{d\mathbf{h}}{d\mathbf{u}^R}\right] \left[\frac{d\mathbf{u}^R}{d\mathbf{u}}\right] \right] \mathbf{p} \tag{15}$$

where $\left[\frac{d\mathbf{h}}{d\mathbf{u}}\right] \in \Re^{m \times m}$ with nonzero diagonal elements, and $\left[\frac{d\mathbf{u}^L}{d\mathbf{u}}\right] \in \Re^{m \times n}$. If we do not incorporate monotonicity enforcement into the reconstruction procedure then a considerable simplification occurs in the calculation of matrix-vector products. The main idea is given in the following almost trivial lemma.

Lemma: Let $v = \mathcal{R}(U) = \mathcal{R}(u_1, u_2, ..., u_n)$ denote an arbitrary order reconstruction operator. If $\mathcal{R}$ depends linearly on u_i then

$$\frac{dv}{du}p = \mathcal{R}(p).$$

Proof: Linearity implies that

$$v = \mathcal{R}(u_1, u_2, ..., u_n) = \sum_{i=1}^{n} \alpha_i u_i$$

so that $\frac{dv}{du_i} = \alpha_i$. The desired result follows immediately

$$\frac{dv}{du}p = \sum_{i=1}^{n} \frac{dv}{du_i} p_i = \sum_{i=1}^{n} \alpha_i p_i = \mathcal{R}(p).$$

This lemma suggests the following procedure for calculation of matrix-vector products, from Eq. 15.

$$\frac{d\mathbf{R}}{d\mathbf{Q}}\mathbf{p} = \mathcal{C}^T \left[\left[\frac{d\mathbf{h}}{d\mathbf{u}^L}\right] \mathcal{R}^L(\mathbf{p}) + \left[\frac{d\mathbf{h}}{d\mathbf{u}^R}\right] \mathcal{R}^R(\mathbf{p}) \right] \tag{16}$$

This amounts to a reconstruction of the vectors $\mathbf{p}^L$ and $\mathbf{p}^R$ from $\mathbf{p}$ using the same reconstruction operator used in the residual computation. Next, the linearized form of the flux function is computed:

$$\mathbf{h}_{lin} = \frac{d\mathbf{h}}{d\mathbf{u}^L}\mathbf{p}^L + \frac{d\mathbf{h}}{d\mathbf{u}^R}\mathbf{p}^R.$$

Finally, the linearized fluxes are assembled using the same procedure as the residual vector assembly. In actual calculations, the conservative flow variables are not reconstructed, thereby necessitating that a change of variable transformation be embedded in the formulation. This is not a serious complication.

4.4 Matrix Preconditioning

In the present applications, we consider a preconditioning matrix based on the incomplete lower-upper (ILU) factorization of the matrix $\mathcal{A}$. ILU preconditioning is a popular and robust preconditioning procedure for use in iterative matrix solvers. ILU factorization is a modification to the standard Gaussian elimination for which the nonzero fill pattern is either preimposed or determined dynamically based on the size or location of fill elements. In this way the amount of storage required can be specified and in some instances minimized. Technical aspects of ILU factorization such as existence and spectral properties have been proven for M-matrices, but the general applicability is much broader and well documented in the literature. The triangular solves required in the application of ILU preconditioning generally give the method global support. This is usually considered a favorable characteristic of the method.

The finite-volume scheme with high order data reconstruction suggests two possible matrices suitable for incomplete factorization.

1. Distance-1 matrix preconditioning. Construct the preconditioning matrix from the Jacobian matrix associated with the lower (first) order accurate discretization of the flow equations. This matrix involves distance-1 neighbors in the triangulation. Matrix-vector products are computed "exactly" using the Jacobian matrix associated with the full second order accurate scheme.

2. Distance-2 matrix preconditioning. Use the Jacobian matrix of the entire second order accurate scheme for both matrix-vector products and preconditioning.

4.5 Performance of GMRES

The viscous multi-element test problem given above provides representative matrices for evaluating the GMRES algorithm. We construct approximate Newton matrices corresponding to flow CFL numbers of 10^3 and 10^8. In addition, distance-1 and distance-2 preconditioning matrices are used to accelerate the algorithms.

Figures 11-12 graph the convergence histories for the GMRES algorithm and the two choices of preconditioner. Since the matrix-vector products and preconditioning solves dominate the iterative calculation, convergence histories are plotted against the number of matrix-vector products required. Each GMRES iteration requires one matrix-vector product. The GMRES algorithm is clearly adversely affected by the distance-1 preconditioning. For this case the distance-1 preconditioned system requires roughly twice as many iterations as the distance-2 preconditioned system. In fact for CFL = 10^8, the convergence is unacceptably slow. In general we find that when using the distance-1 preconditioning matrix, an optimal CFL number exists for convergence and efficiency, which is large but not infinite.

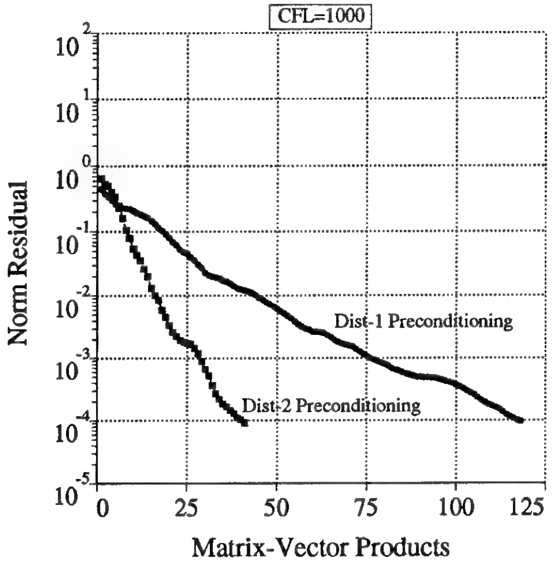


Figure 11: Viscous Flow matrix solution convergence histories for the $GMRES(30)$ algorithm at $CFL = 10^3$ using ILU(0) distance-1 and distance-2 preconditioning matrices.

5 Summary

Practical aspects for Newton-GMRES algorithms from working Navier-Stokes codes have been presented. In particular, implementation issues, such as memory requirements, accuracy requirements for *Ap* products, tradeoffs between full Newton and relax Newton and other pertinent approximations, have been discussed. Two approaches have been highlighted. In the incompressible Navier-Stokes code, the best strategy appears to be an inexact Jacobian (a first order accurate approximation to the third order accurate **Function Evaluation**) for the *Ap* products, a consistent ILU(0) preconditioner, a small subspace size and fairly loose tolerances for **Inner Iteration** convergence. In the unstructured mesh approach, exact *Ap* products are successfully coupled with a first order approximate ILU(0) preconditioner and tighter tolerances levels for **Inner Iteration** convergence. In both cases, an optimal strategy is found producing enhanced efficiencies. Although these conclusions are not universal, they do provide guidelines and practical suggestions for general implementations.

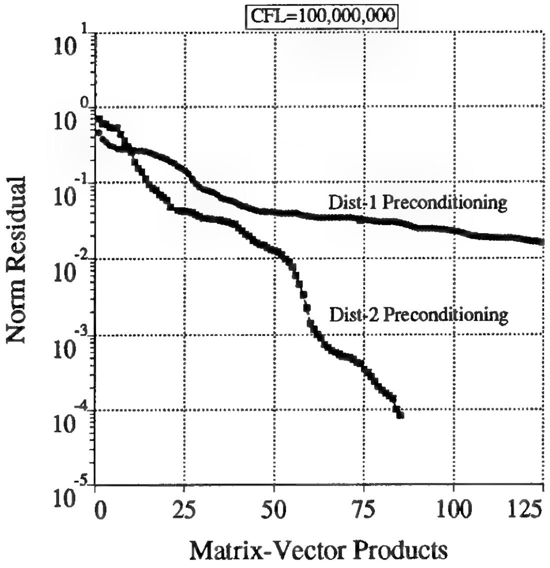


Figure 12: Viscous Flow matrix solution convergence histories for the $GMRES(30)$ algorithm at $CFL = 10^8$ using ILU(0) distance-1 and distance-2 preconditioning matrices.

References

[1] T. J. Barth. Analysis of implicit local linearization techniques for upwind and tvd algorithms. Technical Report AIAA-87-0595, Reno, NV, 1987.

[2] T. J. Barth. *A Three-Dimensional Upwind Euler Solver of Unstructured Meshes.* Technical Report AIAA 91-1548, Honolulu, Hawaii, 1991.

[3] T. J. Barth. *Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-*

Stokes Equations, March 1994. von Karman Institute Lecture Series 1994-05.

[4] T. J. Barth. *Steiner Triangulation for Isotropic and Stretched Elements.* Technical Report AIAA 95-0213, Reno, NV, 1995.

[5] T. J. Barth. An unstructured mesh newton solver for compressible fluid flow and its parallel implementation. Technical Report AIAA-95-0221, Reno, NV, 1995.

[6] T. J. Barth and D. C. Jespersen. *The Design and Application of Upwind Schemes on Unstructured Meshes.* Technical Report AIAA 89-0366, Reno, NV, 1989.

[7] R.M. Beam and H.E. Bailey. Newton's method applied to finite-difference approximations for the steady-state compressiblenavier-stokes equations. *Journal of Computational Physics*, 93(1):108–127, 1991.

[8] P. Brown and Y. Saad. Convergence Theory of Nonlinear Newton-Krylov Algorithms. *SIAM J. Optimization.*, 4:297–330, 1994.

[9] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1):12–26, Aug. 1967.

[10] S. Eisenstat and H. Walker. Globally Convergent Inexact Newton Methods. *SIAM J. Optimization.*, 4:393–422, 1994.

[11] D. C. Jespersen and T. H. Pulliam. Approximate newton methods and flux vector splitting. Technical Report AIAA Paper 83-1899, AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, 1983.

[12] Z. Johan. *Data Parallel Finite Element Techniques for Large-Scale Computational Fluid Dynamics.* PhD thesis, Stanford University, Department of Mechanical Engineering, 1992.

[13] J. A. Meijerink and H. A. van der Vorst. Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems. *Journal of Computational Physics*, 44(1):134–155, 1981.

[14] T. H. Pulliam. Efficient solution methods for the navier-stokes equations. 1985. Lecture Notes for the von Kármán Institute For Fluid Dynamics Lecture Series : Numerical Techniques for Viscous Flow Computation In Turbomachinery Bladings, von Kármán Institute, Rhode-St-Genese, Belgium , 1985.

[15] T. H. Pulliam. *Implicit Methods in CFD.* Clarendon Press, March 1985. The Institute of Mathematics and Its Publications Conference Series,Proceeding of the ICFD 1988 Conference on Numerical Methods for Fluid Dynamics.

[16] P. L. Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *J. Comput. Phys.*, 43, 1981.

[17] S. E. Rogers. A comparison of implicit schemes for the incompressible navier-stokes equations with artificial compressiblity. Technical Report AIAA-95-0567, Reno, NV, 1995.

[18] S. E. Rogers and D. Kwak. An upwind differencing scheme for the time accurate incompressible navier-stokes equations. *AIAA Journal*, 28(2):253–262, Feb 1990.

[19] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.

[20] W. Valarezo, C. Dominik, R. McGhee, and W. Goodman. *High Reynolds Number Confguration Development of a High-Lift Airfoil.* Technical Report AGARD Meeting In High-Left Aerodynamics 10-01, 1992.

[21] V. Venkatakrishnan. Newton solution of inviscid and viscous problems. Technical Report AIAA Paper 88-0413, AIAA 26th Aerospace Sciences Conference, Reno, NV, 1988.

[22] V. Venkatakrishnan. A perspective on unstructured grid flow solvers. Technical Report AIAA Paper 95-0667, AIAA 33rd Aerospace Sciences Conference, Reno, NV, 1995.

[23] L.B. Wigton, N.J. Yu, and D.P. Young. Gmres acceleration of computational fluid dynamics codes. Technical Report AIAA Paper 85-1494, AIAA 7th Computational Fluid Dynamics Conference, Cincinatti, OH, 1985.

HEXAHEDRON BASED GRID ADAPTATION FOR FUTURE LARGE EDDY SIMULATION

J.J.W. van der Vegt and H. van der Ven
National Aerospace Laboratory NLR
P.O. Box 90502, 1006BM Amsterdam, The Netherlands

SUMMARY

This paper discusses a new numerical method which enables the future application of Large Eddy Simulation to high Reynolds number aerodynamic flows. The new numerical method uses local grid refinement of hexahedral cells and the discontinuous Galerkin finite element method. This method offers maximum flexibility in grid adaptation and maintains accuracy on highly irregular grids. The method is demonstrated with calculations of inviscid transonic flow on a generic delta wing. The calculations are done on two parallel shared memory computers and the performance results are used to give estimates of the computing time and memory requirements for a Large Eddy Simulation of a clean wing on a NEC SX-4 supercomputer.

LIST OF SYMBOLS

b_K	external boundary face of element K
$\mathcal{B}$	boundary operator
$C^1[0,T]$	space of one time differentiable functions on the interval $[0,T]$
δ_{ij}	Kronecker delta symbol
E	specific total energy
e_K	face of polyhedron K
$\mathbf{F}^j(\mathbf{U})$	flux vector in Cartesian coordinate direction j
$\hat{\mathbf{F}}(\mathbf{U})$	inner product of $\mathbf{n}^T$ and $\mathcal{F}$
$\mathcal{F}(\mathbf{U})$	matrix with columns $\mathbf{F}^j$
F_K	mapping between elements $\hat{K}$ and K
γ	ratio of specific heats
$\cup_\alpha \Gamma_\alpha$	path in phase space between $\mathbf{U}_h^{int(K)}$ and $\mathbf{U}_h^{ext(K)}$
$\mathbf{h}(\mathbf{U}_h^{int(K)}, \mathbf{U}_h^{ext(K)})$	monotone Lipschitz flux
K	polyhedron element in $\mathcal{T}_h$
K'	neighboring elements of polyhedron K
$\hat{K}$	master element of polyhedron K
$\text{meas}(K)$	measure of polyhedron K
∂K	boundary of polyhedron K
M	maximum number of polynomial terms in expansion of $\mathbf{U}_h$
$[M_K]$	mass matrix of element K
N^+	set of positive natural numbers
$N(K)$	set of neighboring elements of K
$N^\xi(K)$	indices of neighboring elements of K in the ξ-direction
$\mathbf{n}$	unit outward normal vector
Ω	flow domain
$\partial\Omega$	boundary of Ω
p	pressure
$P^k(\hat{K})$	space of polynomial functions of degree $\leq k$ on $\hat{K}$
$P^k(K)$	space of functions whose images under F_K are functions in $P^k(\hat{K})$

Φ_K	limiter function defined on K
Φ_K^i	components of limiter function on K
$\hat{\phi}_j(\xi,\eta,\zeta)$	polynomial basis functions on $\hat{K}$
$\phi_j(\mathbf{x})$	basis functions on K
$\psi_i(\xi,\eta,\zeta)$	trilinear element shape functions
$\mathbf{R}_K$	residual in element K
$R_K^\xi, R_K^\eta, R_K^\zeta$	indicator functions for grid adaptation in ξ, η and ζ directions
R^n	Euclidian n-dimensional space
ρ	density
span	linear span
t	time
T	final time
$\mathcal{T}_h$	triangulation of Ω
$\mathbf{U}$	conservative flow variables
$\bar{\mathbf{U}}_K$	average of $\mathbf{U}$ in element K
$\mathbf{U}_w$	conservative flow variables specified at $\partial\Omega$
$\mathbf{U}_0$	initial conservative flow variables
$\mathbf{U}\|_K$	$\mathbf{U}$ restricted to element K
$\mathbf{U}_h$	numerical approximation of $\mathbf{U}$
$\mathbf{U}^{ext(K)}$	$\mathbf{U}$ at cell face taken as the limit from the exterior of K
$\mathbf{U}^{int(K)}$	$\mathbf{U}$ at cell face taken as the limit from the interior of K
$\mathbf{U}_K^{max}$	maximum $\mathbf{U}$ in K and it's neighboring cells
$\mathbf{U}_K^{min}$	minimum $\mathbf{U}$ in K and it's neighboring cells
$U_{K^*}^i$	components of $\mathbf{U}_h$ at Gauss quadrature points in cell faces of $\hat{K}$
$\hat{\mathbf{U}}_m$	components of polynomial expansion of $\mathbf{U}$ in K
$\tilde{\mathbf{U}}_m$	limited components of polynomial expansion coefficients $\hat{\mathbf{U}}_m$ in K
$\tilde{\mathbf{U}}_h$	limited flow field $\mathbf{U}$ in each element
$\tilde{\mathbf{U}}_K$	vector with limited moments of flow field $\tilde{\mathbf{U}}_m$
u_i	Cartesian velocity components
$\mathbf{V}$	primitive flow variables
$\mathbf{V}_h^k(K)$	vectors with each component $p_i \in P^k(K)$
$\mathbf{W}_h$	vectors which belong to space $\mathbf{V}_h^k$
$\mathbf{x}$	position vector
x_j	components of position vector, $j = \{1,2,3\}$
$\mathbf{x}_K^i$	coordinates of corner points of element K
$\triangle\xi_K$	length of cell in local ξ-direction
ξ,η,ζ	local coordinates in element $\hat{K}$
$\forall$	for all
∇	nabla operator
$\subset$	subset
$\in$	element
$\circ$	composite mapping
$\otimes$	tensor product
^{T}	transposed

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

INTRODUCTION

Computational Fluid Dynamics (CFD) is used for increasingly complicated problems. Many advanced applications of CFD, such as Large Eddy Simulation (LES), can only be done with sophisticated grid adaptation algorithms and require significant computer resources. The aim of this paper is to demonstrate a new grid adaptation algorithm for future application to Large Eddy Simulation. With LES the filtered Navier-Stokes equations are solved which represent the part of the turbulent flow field that can be resolved on the grid. The turbulent length scales which can not be resolved have to be modeled with subgrid scale turbulence models. This approach is quite successful in most parts of the flow field, but as already mentioned by Chapman [3], fails in the near wall region which is critical for LES. Chapman proposed to use successively finer grids close to the wall to capture the viscous sublayer. This reduces the need to model the near wall region where the basic assumption of LES, namely the separation of the flow field in large and small scales, is not valid.

Despite the significant progress made in LES since Chapman's paper the proper solution of the near wall flow field is still one of the key elements preventing LES to be applied to more general problems in aerospace, Moin and Jimenez [10]. The use of successively finer grids can only be done efficiently with sophisticated grid adaptation techniques and requires a numerical scheme which is accurate on highly irregular grids. In this paper a new algorithm is presented, using a combination of local grid refinement and the discontinuous Galerkin (DG) finite element method. This method is capable of efficiently resolving local phenomena such as shear layers and shocks and has the potential to be applied to LES of wall bounded turbulent flows by properly resolving the near wall region. Hexahedron cells are used as basic elements because they suffer less from loss of accuracy due to successive refinements than the more commonly used tetrahedron cells and are more suited to viscous flows. This paper, however, will be limited to inviscid flow in order to demonstrate the basic algorithm.

The discontinuous Galerkin method with Runge-Kutta time integration (RKDG) was originally proposed by Cockburn and Shu [4, 6, 5] for hyperbolic conservation laws. They proved that the RKDG method is TVB stable and satisfies a maximum principle for multi-dimensional scalar hyperbolic conservation laws. This work was mainly theoretical and limited to one and two-dimensional flow fields. The extension to three dimensions was recently presented by van der Vegt [14]. The discontinuous Galerkin method uses a local polynomial expansion in each cell which results in a discontinuity at each cell face. This discontinuity can be represented as a Riemann problem which provides a natural way to introduce upwinding into a finite element method. The DG method can therefore be considered as a mixture of an upwind finite volume method and a finite element method.

A key feature of the DG method is that also equations for the moments of the flow field are solved. In this way a completely local higher order accurate spatial discretization can be obtained without the need to use neighboring cells in the discretization. An alternative to obtain the flow field gradients is to use Gauss' identity, but this method requires grid regularity to be accurate. The use of the moment equations is extremely useful in combination with local grid refinement because no problems with hanging nodes occur and the scheme maintains it's accuracy on highly irregular grids, which generally occur after several grid refinement steps. In this paper the spatial accuracy is limited to second order and the moments represent the flow field gradients. A disadvantage of using the moment equations is that more memory is needed to store the additional moments of the flow field. For future LES applications in wall bounded flows these disadvantages are, however, more than compensated by the increased computational efficiency of the adapted grid.

The DG method makes it easy to mix different types of elements. As basic elements hexahedrons are used, but whenever necessary due to topological degeneracies, prisms, tetrahedrons and other degenerated hexahedrons are used. The initial coarse grid is obtained from a multi-block structured grid, generated with the NLR ENFLOW system. This grid is transformed into an unstructured grid using a face-based data structure, van der Vegt [14]. This data structure is more suited to anisotropic local grid refinement than the commonly used octree data structure. Anisotropic grid refinement is important because many flow phenomena are locally pseudo two-dimensional, eg. shocks and shear layers, and can not be efficiently captured with isotropic grid refinement.

The DG method combined with the face based data structure is extremely local in nature and makes it a good candidate for parallel computing. Parallel computers offer the possibility to overcome the physical limits on single processor speed, but require a significant effort to optimize numerical schemes and coding. LES requires significant computer resources and the performance of the DG method on two different types of parallel shared memory computers, namely a two processor NEC SX-3 and a four processor SGI Power Challenge, will be discussed in this paper. The choice for parallel shared memory computers is made initially to limit the effort in modifying codes.

The outline of the paper is as follows. After a brief description of the governing equations, the DG method will be discussed followed by a description of the grid adaptation algorithm. The algorithm will be demonstrated on the flow field around a generic delta wing. Next, several aspects of using parallel shared memory computers will be discussed and performance results will be presented. These data will be used to give an estimate of the computational complexity of a LES of a clean wing. The papers finishes with concluding remarks.

GOVERNING EQUATIONS

The Euler equations for inviscid gas dynamics in conservation form can be expressed in the flow domain Ω as:

$$\frac{\partial}{\partial t}\mathbf{U}(\mathbf{x},t)+\frac{\partial}{\partial x_j}\mathbf{F}^j(\mathbf{U})=0,$$

Here $\mathbf{x}$ and t represent the coordinate vector, with components $x_i, i=\{1,2,3\}$, in the Cartesian directions, and time, respectively. The Euler equations are supplemented with initial condition $\mathbf{U}(\mathbf{x},0)=\mathbf{U}_0(\mathbf{x})$ and boundary condition $\mathbf{U}(\mathbf{x},t)|_{\partial\Omega}=\mathcal{B}(\mathbf{U},\mathbf{U}_w)$; where B denotes the boundary operator and $\mathbf{U}_w$ the prescribed boundary data. The vectors with conserved flow variables $\mathbf{U}$ and fluxes $\mathbf{F}^j$, $j=\{1,2,3\}$, are defined as:

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_i \\ \rho E \end{pmatrix}; \quad \mathbf{F}^j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p\delta_{ij} \\ u_j(\rho E + p) \end{pmatrix},$$

where ρ, p and E denote the density, pressure and specific total energy and u_i the velocity in the Cartesian coordinate directions x_i, $i = \{1, 2, 3\}$ and δ_{ij} the Kronecker delta symbol. The summation convention is used on repeated indices. This set of equations is completed with the equation of state: $p = (\gamma - 1)\rho(E - \frac{1}{2}u_i u_i)$, with γ the ratio of specific heats.

DISCONTINUOUS GALERKIN APPROXIMATION

The flow domain Ω, which is assumed to be a polyhedron, is covered with a triangulation $\mathcal{T}_h = \{K\}$ of hexahedrons, which are related to the master element $\hat{K}$ through the mapping F_K:

$$F_K : \mathbf{x}(\xi, \eta, \zeta) = \sum_{i=1}^{8} \mathbf{x}_K^i \psi_i(\xi, \eta, \zeta)$$

with $\psi_i(\xi, \eta, \zeta)$ the standard linear finite element shape functions and $\mathbf{x}_K^i$ the coordinates of the vertices of the hexahedron K.

Define on the master element $\hat{K} = [-1, 1]^3$ the space of polynomials: $P^k(\hat{K}) = \text{span}\{\hat{\phi}_j(\xi, \eta, \zeta), j = 0, \cdots, M\}$ and the related space $P^k(K)$ as the space of functions whose images under F_K are functions in $P^k(\hat{K})$: $P^k(K) = \text{span}\{\phi_j(\mathbf{x}) = \hat{\phi}_j \circ F_K^{-1}, j = 0, \cdots, M\}$. In this paper $k = 1$, which yields a second order accurate spatial discretization with polynomials $\hat{\phi} \in \{1, \xi, \eta, \zeta\}$ with $M = 3$.

Define $\mathbf{V}_h^1(K) = \{\mathbf{P}(K) \rightarrow R^5 | p_i \in P^1(K)\}$, then $\mathbf{U}(\mathbf{x}, t) \mid_K$ can be approximated by $\mathbf{U}_h(\mathbf{x}, t) \in \mathbf{V}_h^1(K) \otimes C^1[0, T]$ as:

$$\mathbf{U}_h(\mathbf{x}, t) = \sum_{m=0}^{3} \hat{\mathbf{U}}_m(t)\phi_m(\mathbf{x}). \tag{1}$$

The expansion of $\mathbf{U}$ is local in each element and there is no continuity across element boundaries, which is a major difference with node based Galerkin finite element methods. The element based expansion has as important benefit that hanging nodes, which frequently appear after local grid refinement, do not give any complications. Degenerated hexahedrons, such as prisms and tetrahedrons, which are necessary to deal with topological degeneracies in the grid, are allowed without further complications because the degenerated surfaces do not contribute to the flux balance.

The discontinuous Galerkin finite element formulation of the Euler equations is given by:

Find $\mathbf{U}_h \in \mathbf{V}_h^1(K) \otimes C^1[0, T]$, such that $\mathbf{U}_h(\mathbf{x}, 0) = \mathbf{U}_0(\mathbf{x})|_K \in \mathbf{V}_h^1(K)$, and for $\forall \mathbf{W}_h \in \mathbf{V}_h^1(K)$:

$$\begin{aligned} \frac{\partial}{\partial t} \int_K \mathbf{W}_h^T(\mathbf{x})\mathbf{U}_h(\mathbf{x}, t) d\Omega = \\ - \int_{e_K} \mathbf{W}_h^T(\mathbf{x}) \left(\mathbf{n}^T(\mathbf{x})\mathcal{F}(\mathbf{U}_h)\right) dS \\ - \int_{b_K} \mathbf{W}_h^T(\mathbf{x}) \left(\mathbf{n}^T(\mathbf{x})\mathcal{F}(\mathbf{B}(\mathbf{U}_h, \mathbf{U}_w))\right) dS \\ + \int_K \nabla \mathbf{W}_h^T(\mathbf{x})\mathcal{F}(\mathbf{U}_h) d\Omega, \end{aligned} \tag{2}$$

with $\mathcal{F} = \mathbf{F}^j$, $j = \{1, 2, 3\}$, and $e_K \subset \partial K \backslash \partial\Omega$ and $b_K \subset \partial K \cap \partial\Omega$ the faces of element K in the interior and at the boundary of the domain Ω, respectively. The vector $\mathbf{n}^T$ represents the transposed unit outward normal vector at ∂K.

The flux at the faces e_K, namely $\mathbf{n}^T\mathcal{F}(\mathbf{U}) \equiv \hat{\mathbf{F}}(\mathbf{U})$, is not clearly defined, because the flow field $\mathbf{U}_h$ is discontinuous at the cell faces. The flux is therefore replaced with a monotone flux function $\mathbf{h}(\mathbf{U}_h^{int(K)}, \mathbf{U}_h^{ext(K)})$, which is consistent, $\mathbf{h}(\mathbf{U}, \mathbf{U}) = \hat{\mathbf{F}}(\mathbf{U})$. Here $\mathbf{U}^{int(K)}$ and $\mathbf{U}^{ext(K)}$ denote the value of $\mathbf{U}$ at ∂K taken as the limit from the interior and exterior of K. More details can be found in Cockburn et al. [5]. The use of the monotone Lipschitz flux $\mathbf{h}$ introduces upwinding into the Galerkin method by solving the (approximate) Riemann problem given by $(\mathbf{U}_h^{int(K)}, \mathbf{U}_h^{ext(K)})$. Suitable fluxes are those from Godunov, Roe, Lax-Friedrichs and Osher. In this paper the Osher approximate Riemann solver [11] is used, because of it's good shock capturing capabilities, and the possibility to easily modify the Riemann problem to account for boundary conditions. An important additional reason for the use of the Osher scheme is that it gives an exact solution for a steady contact discontinuity, and therefore it has a very low numerical dissipation in boundary layers, [13], which is important for future extension of the algorithm to the Navier-Stokes equations. The Osher approximate Riemann solver is defined as:

$$\begin{aligned} \mathbf{h}(\mathbf{U}_h^{int(K)}, \mathbf{U}_h^{ext(K)}) &= \frac{1}{2}(\hat{\mathbf{F}}(\mathbf{U}_h^{int(K)}) + \hat{\mathbf{F}}(\mathbf{U}_h^{ext(K)}) - \\ &\quad \sum_\alpha \int_{\Gamma_\alpha} |\partial\hat{\mathbf{F}}| d\Gamma), \end{aligned}$$

where $\cup_\alpha \Gamma_\alpha$ is a path in phase space between $\mathbf{U}_h^{int(K)}$ and $\mathbf{U}_h^{ext(K)}$. Details of the calculation of this path integral in multidimensions can be found in [11]. At the boundary surface the path Γ_α must be modified to account for boundary conditions. In this way a Riemann initial-boundary value problem is solved instead of an initial value problem, [11], and a completely unified and consistent treatment of the flux calculations is obtained, both at interior and exterior faces.

The first order accurate discontinuous Galerkin method with an (approximate) Riemann solver yields monotone results, but second and higher order discretizations need a slope limiter to prevent numerical oscillations around discontinuities and in regions with steep gradients. Cockburn et al. [5] derived a local projection limiter on B-triangulations for multi-dimensional scalar conservation laws, which gives a second order accurate scheme and satisfies a maximum principle when combined with a TVD Runge-Kutta time integration method [12]. The extension to quadrilaterals is presented by Bey and Oden [2], but turned out to be very dissipative.

In this paper a different approach is followed. The second order discontinuous Galerkin method strongly resembles a MUSCL upwind scheme, with as main difference the procedure to determine the flow gradient. In the DG-method the gradient is determined by solving equations for the moments $\hat{\mathbf{U}}_m$, $m = \{1, 2, 3\}$, whereas the MUSCL scheme determines the gradient using data from surrounding cells. The same limiting procedure can, however, be followed. In this paper the multi-dimensional limiter from Barth and Jesperson [1], with the modifications proposed by Venkatakrishnan [15], is used. The limiter from Barth and Jespersen has as benefit that it is a truly multi-dimensional limiter and yields a positive scheme.

The limiter from Barth and Jespersen can, however, seriously degrade convergence to steady state. This was analysed by Venkatakrishnan [15] and the two main causes for this phenomenon are the non-smoothness of the limiter, which uses min- and max-functions, and the fact that the limiter is active in smooth parts of the flow, eg. in the far field.

The limiter according to Venkatakrishnan [15] is directly applied to the conservative variables, which saves the considerable expense of computing the local characteristic decomposition.

Define for each component $\bar{U}_K^i$ of the cell average $\bar{\mathbf{U}}_K = \frac{1}{\text{meas}(K)} \int_K \mathbf{U}_h(\mathbf{x}) d\Omega$:

$$\begin{aligned} U^i_{K\,\text{min}} &= \min_{\forall K' \in N(K)} (\bar{U}^i_K, \bar{U}^i_{K'}) \\ U^i_{K\,\text{max}} &= \max_{\forall K' \in N(K)} (\bar{U}^i_K, \bar{U}^i_{K'}), \end{aligned}$$

with $N(K)$ the set of neighboring cells which connect to cell K. In order to maintain monotonicity the approximate flow field $\mathbf{U}_h$ must satisfy $\mathbf{U}_h(\mathbf{x}) \in [\mathbf{U}_K^{min}, \mathbf{U}_K^{max}], \forall \mathbf{x} \in K$, which is accomplished with the limiter function $\mathbf{\Phi}_K$ defined as:

$$\mathbf{\Phi}^i_K = \begin{cases} \phi_L\left(\frac{U^i_{K\,\text{max}} - \bar{U}^i_K}{U^i_{K*} - \bar{U}^i_K}\right) & \text{if} \quad U^i_{K*} - \bar{U}^i_K > 0 \\ \phi_L\left(\frac{U^i_{K\,\text{min}} - \bar{U}^i_K}{U^i_{K*} - \bar{U}^i_K}\right) & \text{if} \quad U^i_{K*} - \bar{U}^i_K < 0 \\ 1 & \text{if} \quad U^i_{K*} - \bar{U}^i_K = 0 \end{cases}$$

Here U^i_{K*} are the components of $\mathbf{U}_h$ at the Gauss quadrature points in $\hat{K}$, used to evaluate the integrals in equation (2). The function $\phi_L(y)$ replaces $\min(1, y)$ in the original Barth and Jesperson limiter and is defined as:

$$\phi_L(y) = \frac{y^2 + 2y}{y^2 + y + 2}$$

Defining $\Delta = U^i_{K*} - \bar{U}^i_K$, $\Delta_+ = U^i_{K\,\text{max}} - \bar{U}^i_K$ and $\Delta_- = U^i_{K\,\text{min}} - \bar{U}^i_K$ and replacing $\Delta^2_\pm$ with $\Delta^2_\pm + \epsilon^2$ a smoother limiter is obtained:

$$\mathbf{\Phi}^i_K = \begin{cases} \frac{\Delta^2_+ + \epsilon^2_K + 2\Delta\Delta_+}{\Delta^2_+ + \epsilon^2_K + 2\Delta^2 + \Delta\Delta_+} & \text{if} \quad \Delta > 0 \\ \frac{\Delta^2_- + \epsilon^2_K + 2\Delta\Delta_-}{\Delta^2_- + \epsilon^2_K + 2\Delta^2 + \Delta\Delta_-} & \text{if} \quad \Delta < 0 \\ 1 & \text{if} \quad \Delta = 0 \end{cases}$$

The coefficient ϵ_K is set equal to $\epsilon_K = (C\Delta s_K)^3$, with Δs_K the minimum distance between the cell face centers of two opposite faces of element K. The constant C determines the balance between limiting and no limiting and thereby influences the convergence to steady state. If $C = 0$ the original Barth and Jespersen limiter is obtained. In this paper $C = 1$ is used.

The limiter $\mathbf{\Phi}_K$ is applied independently to each component of the flow field: $\tilde{U}^i_m = \Phi^i_K \hat{U}^i_m$, $m = \{1, 2, 3\}$. This is slightly less robust then using $\mathbf{\Phi}_K = \min_i \Phi^i_K$, but gives significantly less numerical dissipation. The coefficients $\hat{\mathbf{U}}_m, m = \{1, 2, 3\}$ in equation (1) represent the gradient of the flow field with respect to the local coordinates in $\hat{K}$. This modification of the local gradient would violate conservation of $\mathbf{U}$ in K, which can be corrected by modifying the coefficient $\hat{\mathbf{U}}_0$:

$$\tilde{U}^i_0 = \hat{U}^i_0 + \frac{1 - \Phi^i_K}{\text{meas}(K)} \sum_{m=1}^{3} \hat{U}^i_m \int_K \phi_m(\mathbf{x}) d\Omega$$

This relation is obtained from the condition $\frac{1}{\text{meas}(K)} \int_K \tilde{\mathbf{U}}_h(\mathbf{x}) d\Omega = \bar{\mathbf{U}}_K$. The limited flow field in cell K is then equal to:

$$\tilde{\mathbf{U}}_h(\mathbf{x}, t) = \sum_{m=0}^{3} \tilde{\mathbf{U}}_m(t) \phi_m(\mathbf{x}).$$

The final discontinuous Galerkin finite element discretization is now obtained by evaluating the integrals over the element K and it's boundary ∂K in equation (2). This is done using the transformation F_K, between K and the master element $\hat{K}$. The integrals $\int \mathbf{W}_h^T \mathbf{U}_h d\Omega$, are calculated analytically, which requires quite some algebra, whereas the other integrals are calculated with Gauss quadrature rules. Cockburn et al. [5] proved that if the quadrature rules for the surface integrals in equation (2) are exact for polynomials of degree $(2k + 1)$ and exact for polynomials of degree $2k$ for the volume integrals then the spatial accuracy of the DG method is $k + 1$. In order to preserve uniform flow it is necessary to use quadrature rules which are exact for polynomials of order 3. For $k = 1$ the surface integrals are calculated with four point Gauss quadrature rules. The volume integrals require six point Gauss quadrature rules.

The use of four and six point Gauss quadrature rules is, however, unnecessarily expensive. The number of flux calculations in the approximation of the surface integrals can be reduced from four to one using the following approximation, which is second order accurate in the mean:

$$\begin{aligned} \int_{\partial\Omega} \phi_n(\mathbf{x}) \mathbf{n}^T \mathcal{F}(\mathbf{U}) d\Omega &= \int_{\partial\hat{\Omega}} \hat{\phi}_n(\mathbf{x}) \mathbf{n}^T \mathcal{F}(\mathbf{U}) J_e d\hat{\Omega} \\ &\cong \mathcal{F}(\mathbf{U})|_c \int_{\partial\hat{\Omega}} \hat{\phi}_n(\mathbf{x}) \mathbf{n}^T J_e d\hat{\Omega} \end{aligned}$$

with $\mathcal{F}(\mathbf{U})|_c$ calculated at the cell face center and J_e the Jacobian of the transformation of the cell face $\partial\Omega$ to $\partial\hat{\Omega}$ on the master element $\hat{K}$. The integrals $\int_{\partial\hat{\Omega}} \hat{\phi}_n(\mathbf{x}) \mathbf{n}^T J_e d\hat{\Omega}$ are precalculated with four point Gauss quadrature rules, which are exact using elements defined with linear shape functions, and therefore free stream consistency is preserved with this approximation. A similar approximation can be made for the volume integral $\int_K \nabla \mathbf{W}_h^T(\mathbf{x}) \mathcal{F}(\mathbf{U}_h) d\Omega$, with $\mathcal{F}(\mathbf{U})$ calculated in the center of $\hat{K}$ and the geometrical part of the volume integral precalculated with a six point Gauss quadrature rule. This formulation requires about four times less computing time than using the more accurate evaluation of the flux integrals and yields similar results. The discretization using four and six Gauss quadrature points for the surface and volume integrals yields, however, a slightly more robust scheme on coarse grids. This is mainly due to the fact that the cross-coupling terms in the moment equations are retained in this case.

For each element K a system of ordinary differential equations is now obtained:

$$[M_K] \frac{\partial}{\partial t} \tilde{\mathbf{U}}_K = \mathbf{R}_K$$

with $\tilde{\mathbf{U}}_K$ a vector with the moments of the flow field in each element, $\tilde{\mathbf{U}}_m, m = \{0, \cdots, 3\}$, and $\mathbf{R}_K$ the right-hand side of equation (2). The equations for $\frac{\partial}{\partial t}\tilde{\mathbf{U}}_K$ are integrated in time using the third order TVD Runge-Kutta scheme from Shu [12]. For steady state calculations convergence is accelerated using local time stepping.

A significant difference with node based FEM is that the mass matrix $[M_K]$ is uncoupled for each element K and can be easily inverted.

DIRECTIONAL GRID ADAPTATION

The use of increasingly finer grids in LES in the near wall region, as proposed by Chapman [3], and in other regions with strong shear layers or shocks can be most efficiently done using local grid refinement. The grid is locally enriched by subdividing cells, independently in each of the three local grid directions, ξ, η or ζ, of $\hat{K}$. This anisotropic grid refinement is more efficient in capturing local flow phenomena than isotropic refinement, because many flow features are frequently pseudo two-dimensional. A coarse initial grid is used, which is generated with a multi-block structured grid generator, and transferred into an unstructured hexahedron grid. If necessary degenerated hexahedrons, such as prisms and tetrahedrons, are allowed to deal with topological degeneracies. After calculating the flow field, the grid cells are split in the local ξ-direction if:

$$\frac{R_K^\xi}{\max_{\forall K \in \mathcal{T}_h} R_K^\xi} > \text{tolerance}$$

with the sensor function R_K^ξ for the cell K defined as:

$$R_K^\xi = \max_{i \in \{1,\cdots,5\}, \forall K' \in N^\xi(K)} (V_K^i - V_{K'}^i)^2 \triangle \xi_K^2 \qquad (3)$$

Here $\triangle \xi_K$ is the length of cell K in the local ξ-direction, $\mathbf{V} = (\rho, u, v, w, p)^T$ the vector with primitive variables and $N^\xi(K)$ the indices of the neighboring cells of cell K in the ξ-direction. Equivalent expressions are used for the η and ζ directions. This sensor is based on an equidistribution principle, see for instance Marchant et al. [9]. An important advantage of this sensor is that it prevents regions with discontinuities from constantly dominating the local grid refinement. After several refinements the relative contribution of regions with discontinuities reduces, because $\triangle \xi_K$ in equation 3 becomes progressively smaller.

DATA STRUCTURE

The discontinuous Galerkin method with local grid refinement of hexahedrons requires a significantly different data structure than the frequently used edge based data structure. The edge based data structure is very efficient for unstructured vertex based schemes using tetrahedrons. The discontinuous Galerkin method is a cell based algorithm and the primary calculations are the evaluation of fluxes through cell faces. This can be done efficiently using a face based data structure. A face based data structure also has as important benefit that there are no limitations on the number of cells which can connect to one cell face and is crucial for local grid refinement. The alternative would be an octree data structure, but this data structure does not combine well with anisotropic grid refinement. In van der Vegt [14] an algorithm is presented to determine all face to cell connections efficiently. The main element in this algorithm is that cell faces are split into smaller subfaces until each face connects only to one cell on each side. There are no limits on the number of neighboring cells and using advanced searching algorithms a very efficient scheme is obtained, which can establish all face to cell connections in $O(N \log_2(N))$ operations with N the number of faces. The fluxes are calculated in one loop over all the faces, which can be fully vectorized using a coloring scheme. The face based data structure does not put any limitations on the number of neighboring cells, but if the number of cells connecting to one face becomes too large then the number of colors significantly increases. This reduces the efficiency on vector and parallel computers and will be a topic of future research. In the grid adaptation process cells are added and deleted which is done efficiently using AVL-trees, for more details see van der Vegt [14]

Adaptation Step	Cells	Grid Points	Faces
0	19152	20790	59594
1	33094	38277	132038
2	49088	63357	203400
3	73091	104435	307783
4	124030	197424	538109
5	211578	357752	933616
6	322708	592441	1447763

Table 1: *Number of cells, grid points and faces after each adaptation step*

DISCUSSION AND RESULTS

The grid adaptation algorithm has been tested on the flow around a generic delta wing. The geometry is a cropped-delta wing with a 65-degree sweep angle and a sharp leading edge. A constant airfoil section in the streamwise direction is used (modified NACA 64A005 profile; straight line aft of 75% chord) with 5% relative thickness, no twist and camber. More information about the geometry and experimental results can be found in Elsenaar et al. [7]. A transonic flow test case is used with angle of attack $\alpha = 20°$ and free stream Mach number $M_\infty = 0.85$. The initial grid consisted of 19152 cells and 20790 grid points. The grid is adapted six times, independently in all three directions and the final grid consists of 322708 cells and 592441 grid points, see Table 1. During each adaptation step approximately 15 % of the cells is deleted, after which the number of cells is increased between 70 % and 90%. The removal of grid cells is important, because initially on the coarse grid the refinement sensor is less accurate and some unnecessary refinement takes place. Local time stepping is used and significantly improves convergence to steady state, see Figure 1. The sharp peaks in the convergence plot are caused by the grid adaptation, except for the first peak, which results from freezing the slope limiter after 750 time steps to improve converge. Freezing of the slope limiter is not necessary after grid adaptation.

Figure 3 shows the pressure field and grid lines on the leeward side of the delta wing. The flow field is dominated by a strong primary vortex which starts at the apex and moves downstream under an angle of 20 degrees with the streamwise direction. Vorticity is generated at the sharp leading edge in a thin vortex sheet and rolls-up into the primary vortex. The velocity under this vortex, just above the upper surface, becomes very large and a strong shock develops between the primary vortex and

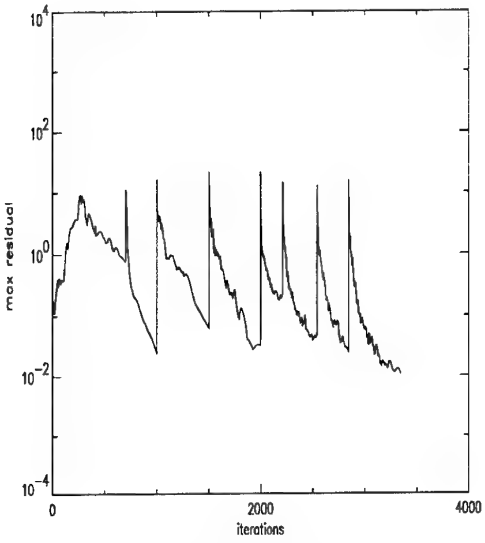


Figure 1: *Maximum residual in flow field.*

the upper surface, see also Figures 4 and 6. The benefits of anisotropic grid refinement are very clear in Figure 3, where the grid is strongly adapted along the primary vortex in the first 85% of the delta wing, where the flow field is approximately conical. At a chord length of 85%, where the sharp leading edge connects to the tip, the primary vortex and related shock have a sharp kink, see Figures 3 and 6. Two shocks develop in the primary vortex. One normal to the leading edge and connected to the kink in the shock structure under the primary vortex and another one from the same location on the leading edge and connected more upstream to the shock under the primary vortex. A similar shock structure, although slightly more downstream, was observed by Hoeijmakers et al. [8] using a much finer structured grid. This shock structure has a strong influence on the primary vortex, which completely blows up behind it, see Figure 5, and is very well captured by the grid adaptation. Also visible in Figure 5 is that the grid is adapted to the trailing edge vortex. The primary vortex significantly grows after 85% chord and merges with the tip vortex, see Figure 4. Also visible is the start of roll-up of the wake, which develops into a mushroom type vortex structure. In addition to the shock structures in and around the primary vortex there is also a shock starting at about 75% downstream at the center line and connected to the trailing edge at approximately mid span. A better view of this shock can be obtained in Figure 6 which gives a perspective view of the delta wing and the grid and flow field at approximately 70% chord. Figure 5 clearly shows the strong primary vortex and the shock between the vortex and body. Also visible is the significant refinement in this region and the vortex layer starting at the sharp leading edge.

PARALLELIZATION

The above described algorithm has been implemented in the program Hexadap, which is parallelized on shared memory machines, namely:

- A two processor NEC SX-3/22 with a peak performance of 2 × 2.75 GFlop/s, a main memory unit (MMU) of 1 GByte and 4 GByte Extended Memory Unit (XMU) of which 1.2 GByte can be efficiently used to store run-time data,
- A four processor SGI Power Challenge with a peak performance of 4 × 350 MFlop/s, main memory of 256 MByte and 16 KByte primary and 4 MByte secondary cache.

	Small	Medium	Large	Long
vector length	8000	2000	1000	120,000
iterations	100	100	300	100
adaptations	0	1	1	0

Table 2: *Problem sizes*

The parallelization uses microtasking, adding parallelization compiler directives, for both machines, and macrotasking, explicitly assigning tasks to different processors. (Implementation on the SGI Power Challenge is done with the CONCURRENT CALL assertion). The advantage of microtasking is that the code remains portable. The advantage of macrotaking is that large tasks can be assigned, even if the tasks have no do-loop structure, and memory can be used more efficiently.

The above described algorithm consists of two parts, namely grid adaptation and flow computation. The grid adaptation part, which consists predominantly of scalar operations, requires a domain decomposition for parallelization and is not considered in this paper. The flow computation has as most important component the calculation of cell face fluxes and consists of loops over the cell faces. The result is added to the residual in the two cells connected to each cell face. The loops use indirect addressing and in order to vectorize these loops a coloring scheme has been applied.

The initial flow field and the flow field after three and six adaptations is used to test the parallel performance of the flow solution algorithm, see Table 1. These cases are denoted Small, Medium and Large. The average vector length and number of iterations are presented in Table 2, which shows that the average vector length decreases with problem size. This is caused by the increasing number of colors after grid adaptation. A reduction in the number of colors is possible by limiting the number of neighboring cells connected to each cell face. In order to investigate the dependence of the performance results on the vector length a special case, labeled Long, is also tested, see Table 2.

NEC SX-3

The two computationally most intensive parts of the flow solution algorithm are the routines Limit and Flux. Limit applies a slope limiter to ensure monotonicity and Flux computes the fluxes through cell faces. The suffixes 1G and 4G in Tables 3 and 4 refer to the number of Gauss quadrature points used in the evaluation of the flux integrals. The two routines constitute 90% of the total computing time. They have roughly the same structure: a nested loop, first over all colors and then over all faces of one color.

MFlop rates on a single processor NEC SX-3 are reported in Table 3. The rates are based on flop counts and elapsed times. The decrease in overall performance for the Medium and Large problems is caused by the larger number of colors after grid adaptation which results in a reduced vector length. The case Long does not suffer from this reduction in performance. Also indicated in Table 2 is if the grid is adapted.

	Flux	Limit	Total
Small(4G)	474	392	426
Medium(1G)	406	258	232
Large(1G)	371	241	265
Long(4G)	484	445	452
Long(1G)	463	318	314

Table 3: *Megaflop rates on single processor NEC SX-3 (based on elapsed times)*

Two parallelization strategies have been tested. The first strategy executes the loops over the colors in parallel and vectorizes the inner loop over the faces. Part of the inner loop over the faces consists of an update of the residuals at the cell centers. Within one color all faces connect to cells with different cell addresses, but this is not assured between different colors, causing a data dependency. Hence, in the above parallelization strategy, the residual updates have to be performed in a critical section, where only one processor is active at a time. The second strategy divides the loop over the faces within one color over the available processors. The main problem with this approach is that sufficient vector length should remain after loop division.

The MFlop rates and speedup results are presented in Table 4. The timings and speedups are influenced by the use of the external memory unit XMU of the SX-3. The XMU allows for fast access to data which cannot be placed in core memory. Sequentially, the use of the XMU instead of core memory hardly decreases performance. During parallel execution, however, locks applied during I/O seriously deterioriate the performance. If we compensate for the time spent during I/O to the XMU speedups increase, the corrected speedups are labeled Corr in Table 4. The MFlop rates in Table 4 are based on the corrected speedups.

The results for the first parallelization strategy, namely parallel execution of loops over the colors, are obtained using micro tasking and are labeled 'C' in Table 4. The speedups are with respect to elapsed times. It is clear from the results that the efficiency of the parallelization is rather low. This has two reasons. First, the critical section consumes 20% of the computing time, and second, the parallel system overhead is about 10%. This large sequential part limits the maximum attainable speedup on more processors to 5.

The second parallelization strategy, namely parallel execution of loops over the faces within one color, does not suffer from a critical section. At first the code was parallelized using microtasking. The program structure is such that the flux computation is split into many different loops in different functional subroutines. Therefore the computational load per loop is low, less than 1.5 msec. It turned out that this load is too low to be efficient on the NEC SX-3: the parallel overhead was as large as, or even larger than the parallel gain and no speedup was obtained.

Using macrotasking the parallel overhead could be reduced significantly. Instead of parallelizing each loop separately, the work is divided into two tasks in the subroutines Flux and Limit, each task doing the same job as the subroutines, but on only half the loop. This not only reduced the parallel system overhead, but also reduced memory use. In microtasking local data is copied for each processor, in macrotasking the local data can be defined per task, and thus approximately halved with respect to the sequential program. Memory use for the medium sized problem is 498 MByte for the sequential program, 540 MByte for the microtasked program and 515 MByte for the macrotasked program. Speedups for the macrotasked program are presented in Table 4 and labeled 'F'.

		Flux	Corr	Limit	Total	Corr	MFlop/s
SX-3	C	1.5	1.6	1.6	1.4	1.5	624
Small(4G)	F	1.6	1.6	1.7	1.3	1.3	566
SX-3	C	1.5	1.8	1.2	1.5	1.6	364
Medium(1G)	F	1.3	1.6	1.4	1.5	1.6	376
SX-3	C	1.4	1.8	1.2	1.2	1.3	356
Large(1G)	F	1.1	1.4	1.2	1.1	1.2	322
SX-3	C	1.5	1.5	1.4	1.3	1.4	614
Long(4G)	F	1.7	1.7	1.6	1.5	1.6	701
SX-3	C	1.5	1.8	1.3	1.3	1.4	440
Long(1G)	F	1.6	1.9	1.6	1.5	1.6	495
SGI	LL	2.9	-	3.3	2.1	-	85
Small(4G)	F	3.7	-	2.9	2.3	-	94
SGI	LL	1.5	-	2.0	1.5	-	37
Medium(1G)	F	2.9	-	2.3	2.0	-	51

Table 4: *Speedups relative to single processor performance (based on elapsed times); SX-3 two processors; SGI four processors; C: parallel loop over colors (microtasking); F: parallel loop over faces within one color (macrotasking); LL: Low level microtasking*

The decrease in parallel performance with increased problem size can be attributed to the reduced vector length. This is clearly demonstrated by the results of test case Long, which has an average vector length of 120000 in the loops over the cell faces. This problem reaches the highest parallel performance, with a speed-up of 1.9 in routine Flux. Another factor which significantly reduces the performance of the flow solution algorithm on a NEC SX-3 computer is the limited memory bandwidth. This is especially important for the large number of indirectly addressed loops and a main reason for the big gap between sustained and peak performance. The memory bandwidth limitations are the most evident in Limit, where the ratio between computations and load/stores is rather low.

SGI Power Challenge
The SGI Power Challenge has scalar processors and therefore no problems with data dependencies within a processor. The code was therefore parallelized using the second parallelization strategy, namely parallel execution of the loops over the cell faces. Only the Small and Medium problems were tested, since the other problems did not fit in memory.

Two implementations are made, one by parallelizing each loop separately (low-level), and one using the same macrotasking structure as described in the previous section. Parallelization is straightforward using the parallel code of the SX-3. Directives are changed to SGI directives. The macrotasking is accomplished using the CONCURRENT CALL assertion.

Results of speedups and MFlop rates are presented in Table 4. The low-level parallelization is labeled 'LL' and the macrotask-

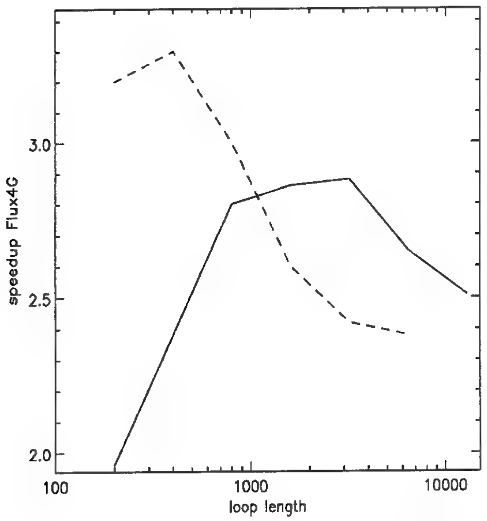


Figure 2: *Cache dependency of speedups on the SGI Power Challenge in routine Flux(4G) (— Small - - - - Medium)*

ing results are labeled 'F'. Since the SGI has no XMU there is no correction for the speedups: the entire program is run in core memory. The speedups for macrotasking are better than for the low level parallelization. The performance in MFlops of the SGI four processor Power Challenge, as listed in Table 4, is between 10% and 17% of the two processor SX-3 performance and not sufficient for large scale computing. The percentage of peak performance is between 3% and 7% on the SGI Power Challenge and between 6% and 13% on the two processor NEC SX-3.

Results of the SGI Power Challenge are rather sensitive to cache misses. A parameter in the flow solution algorithm determines the number of cell faces in the flux calculation processed at one time. Varying this parameter changes the amount of the data being processed, and can be used to optimize the cache use of the program. Significant differences can occur, and the optimal value of the parameter depends on the problem at hand. (see Figure 2). The speedups of Table 4 are computed using the optimal timing results.

Estimate of the computing time for a LES of a clean wing on a NEC SX-4/16 computer

The parallel performance on the NLR NEC SX-3/22 has been used to estimate the problem size of a Large Eddy Simulation of a clean wing on a 16 processor NEC SX-4, which will be delivered to NLR in 1996. The NLR NEC SX-4/16 is expected to have a peak performance of 32 Gflop/s, a main memory of 4 GByte and 8 GByte XMU. With respect to the SX-3/22 its architecture is more suited for indirect addressing and a single processor speedup of 2 is expected for programs using indirect addressing.

The size of the LES is primarily determined by the available memory. Let N be the number of grid points, and n the number of flow variables. For a Large Eddy Simulation with a one-equation turbulence model we have $n = 6$. The memory use of Hexadap is $8(12n + 40)N + 2 \cdot 10^8$ Byte. With an available memory of 8 Gbyte and 8 bytes per variable the maximum number of grid points $N = 9 \cdot 10^6$. Using the estimates given by Chapman [3], this number allows for a LES with sublayer resolution around a clean wing at a Reynolds number of approximately 10^6.

The computing time for one time step is estimated from the relation:

$$t_{cpu} = \frac{n \cdot N}{S_A \cdot S_C} \cdot \left(\frac{f_S}{r_S} + \frac{1}{S_{16}} \left(\frac{1.3 \cdot 1.1 \cdot f_F}{r_F} + \frac{f_L}{r_L} + \frac{f_R}{r_R} \right) \right)$$

with: S_A a factor to account for grid adaptation, $S_A = 0.9$, S_C the single processor speedup of the NEC SX-4 compared to the SX-3, $S_C = 2$. The suffixes S, F, L and R refer to the following parts of the algorithm: S, serial part, F, subroutine Flux(1G), L, subroutine Limit, and R the remaining part of the flow solution algorithm which is parallelizable. The variables f_* denote flop counts in the respective parts of the algorithm to advance one flow variable one time step in one grid point. The measured values are: $f_S = 90$, $f_F = 1570$, $f_L = 880$ and $f_R = 180$. The variables r_* denote the measured flop rates in the respective parts of the algorithm and are equal to: $r_S = r_R = 350 \cdot 10^6$, $r_F = 463 \cdot 10^6$ and $r_L = 350 \cdot 10^6$ flop/s. The flop count in routine Flux is increased with 10% for the viscous contribution and 30% for a one-equation subgrid model using the Germano approach. The parallel speedup, denoted by S_{16} on a 16 processor NEC SX-4 is estimated as twelve. The computing time required to advance one time step on a grid with $9 \cdot 10^6$ grid points is then approximately 28 seconds.

The time scale of the smallest eddies in the flow field will be approximately 100 times larger than the CFL limit for an explicit scheme. The CFL time step limitation can be removed with an implicit, time accurate temporal discretization using multigrid acceleration. With these assumptions a Large Eddy Simulation of a clean wing at a Reynolds number 10^6 on a mesh with $9 \cdot 10^6$ grid points which evolves 6500 time steps, which should be sufficient to obtain a reasonable statistical sample, would require 50 hours on a 16 processor NEC SX-4.

Conclusions of the parallelization

Provided that the vector length is sufficient, the most efficient parallelization strategy for the present flow solution algorithm is a high level parallelization of loops over faces of one color using macrotasking. Macrotasking reduces parallel system overhead and memory use. Correcting for the XMU a maximum speedup of 1.9 is reached on a two processor SX-3.

There are three causes for the not perfect overall performance on the NEC SX-3:

- I/O between Main Memory and XMU in parallel processing takes significantly more time,
- Vector length decreases, and hence single processor speed,
- Parallel system overhead.

Concerning the latter cause, the balance between the two processors is, when corrected for the I/O between MMU and XMU, as predicted by the size of the parallel part of the algorithm. Hence, the computational load is well balanced, and the remaining performance loss can only be explained by parallel system overhead. Since the NEC SX-3 is not primarily suited for parallel use, the relatively high parallel system overhead is not too surprising. It is expected that the NEC SX-4 has significantly less overhead.

Low-level do-loop parallelization on the NEC SX-3 turns out to be only sufficient for loops with a computational load greater than 1.5 msec.

The parallel efficiency on the SGI Power Challenge is similar, the percentage of peak performance is relatively low, even compared with the NEC SX-3. Moreover, the cache sensitivity makes the optimization problem dependent.

The present parallelization on the NEC SX-3 will not be sufficiently efficient on the 16 processor SX-4. The parallel execution of the loops over the cell faces is inefficient since the loop length will be too short to be divided over 16 processors. This problem can be solved by limiting the number of neighboring cells connected to one cell face to at most four, which significantly reduces the number of colors and thereby increases vector length. The parallel execution of the loop over the colors contains a sequential part of 20%, and hence has a maximum speedup of 5. This sequential part can be eliminated using a domain decomposition of the grid, which also has as main benefit that the grid adaptation part can be executed in parallel.

CONCLUDING REMARKS

The discontinuous Galerkin finite element method with local grid enrichment has been demonstrated on the three-dimensional, inviscid flow field around a delta wing at transonic speed. The use of anisotropic grid refinement of hexahedron type cells is effective in capturing the shock structure and primary vortex on the leeward side of the delta wing. The discontinuous Galerkin method works well on highly irregular grids and is therefore a good candidate for Large Eddy Simulations, because it offers the opportunity to capture viscous sublayers with successively finer grids through local grid refinement. An estimate of the required computational resources for such a simulation is presented. The use of a face based data structure works well in combination with local grid refinement and allows efficient vectorization and parallelization of the code. On the NEC SX-3 the possible speedup through parallelization strongly depends on the vector length. A maximum speed-up of 1.9 on the two processor NEC SX-3 is obtained when sufficient vector length was available. A good parallel performance, with a speed-up of 3.7, is obtained on the four processor SGI Power Challenge, but the results are sensitive to cache misses.

From the present results it is estimated that for future LES applications in wall bounded flows, the gain from the increased computational efficiency obtained from highly adapted grids more than compensates the increased number of operations and memory use. A LES of a clean wing at a Reynolds number of 10^6 will become feasible on a 16 processor NEC SX-4 in a turnaround time of one weekend. Significant further developments, such as the addition of the viscous contribution and implicit time-accurate temporal discretization using multigrid acceleration (in progress), will, however, be needed to reach this goal.

REFERENCES

[1] Barth, T.J. and Jespersen, D.C. The design and application of upwind schemes on unstructured meshes. AIAA Paper 89-0366, 1989.

[2] Bey, K.S. and Oden, J.T. A Runge-Kutta discontinuous finite element method for high speed flows. AIAA Paper 91-1575-CP, 1991.

[3] Chapman, D.R. Computational aerodynamics development and outlook. AIAA Paper 79-0129, 1979.

[4] Cockburn, B and Shu, C.W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Math. Comp.*, 52:411–435, 1989.

[5] Cockburn, B., Hou, S. and Shu, C.W. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comp.*, 54:545–581, 1990.

[6] Cockburn, B., Lin, S.Y. and Shu, C.W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *JCP*, 84:90–113, 1989.

[7] Elsenaar, A., Hjelmberg, L., Bütefisch, K.A. and Bannink, W.J. The international vortex flow experiment. AGARD Symposium on Validation of Computational Fluid Dynamics, Lisbon, AGARD CP 437, 1987, also AGARD Advisory Report 303, 1994.

[8] Hoeijmakers, H.W.M., Jacobs, J.M.J.W. and Van Den Berg, J.I. Numerical simulation of vortical flow over a delta wing at subsonic and transonic speed. Presented at 17th ICAS Congress, 1990, Stockholm,Sweden, 1990.

[9] Marchant, M.J. and Weatherhill, N.P. Adaptivity techniches for compressible inviscid flows. *Comp. Meth. in Appl. Mech. and Eng.*, 106:83–106, 1993.

[10] Moin, P. and Jimenez, J. Large eddy simulation of complex turbulent flows. AIAA Paper 93-3099, 1993.

[11] Osher, S. and Chakravarthy, S. Upwind schemes and boundary conditions with applications to Euler equations in general geometries. *JCP*, 50:447–481, 1983.

[12] Shu, C.W. and Osher, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *JCP*, 77:439–471, 1988.

[13] Van Der Vegt, J.J.W. Higher-order accurate Osher schemes with application to compressible boundary layer stability. AIAA Paper 93-3051, 1993.

[14] Van Der Vegt, J.J.W. Anisotropic grid refinement using an unstructured discontinuous Galerkin method for the three-dimensional Euler equations of gas dynamics. AIAA Paper 95-1657, 1995.

[15] Venkatakrishnan, V. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *JCP*, 118:120–130, 1995.

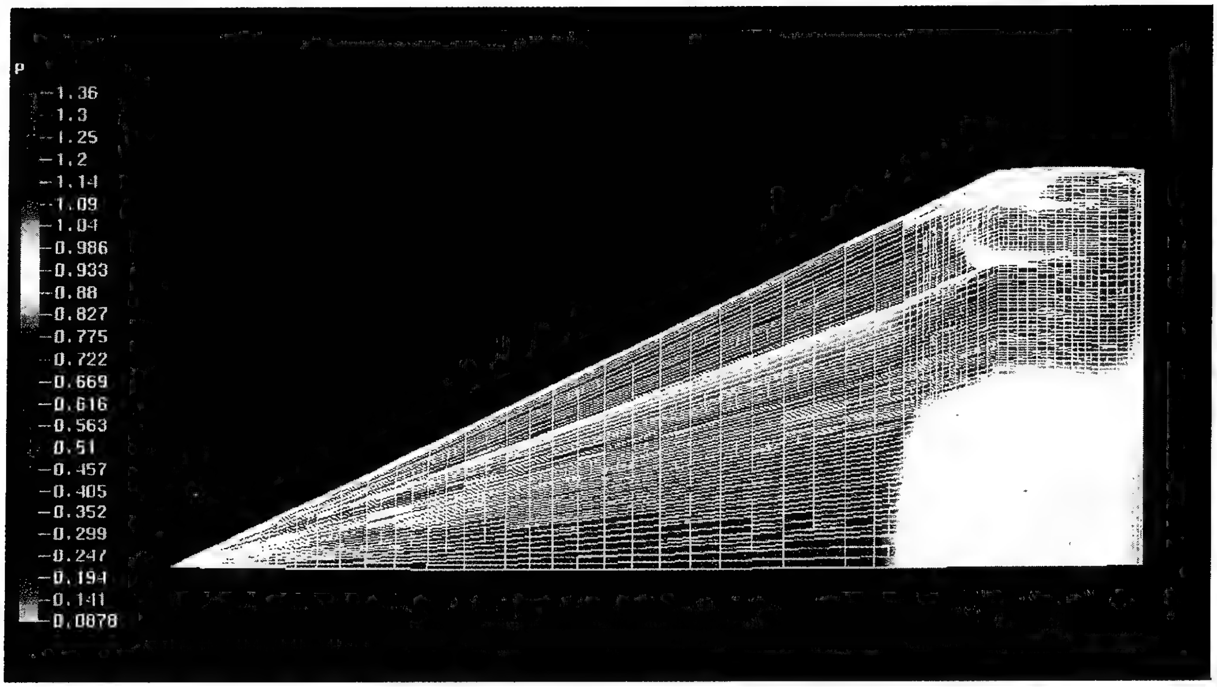


Figure 3. *Pressure field and adapted grid on upper surface of delta wing. ($M_\infty = 0.85, \alpha = 20°$)*

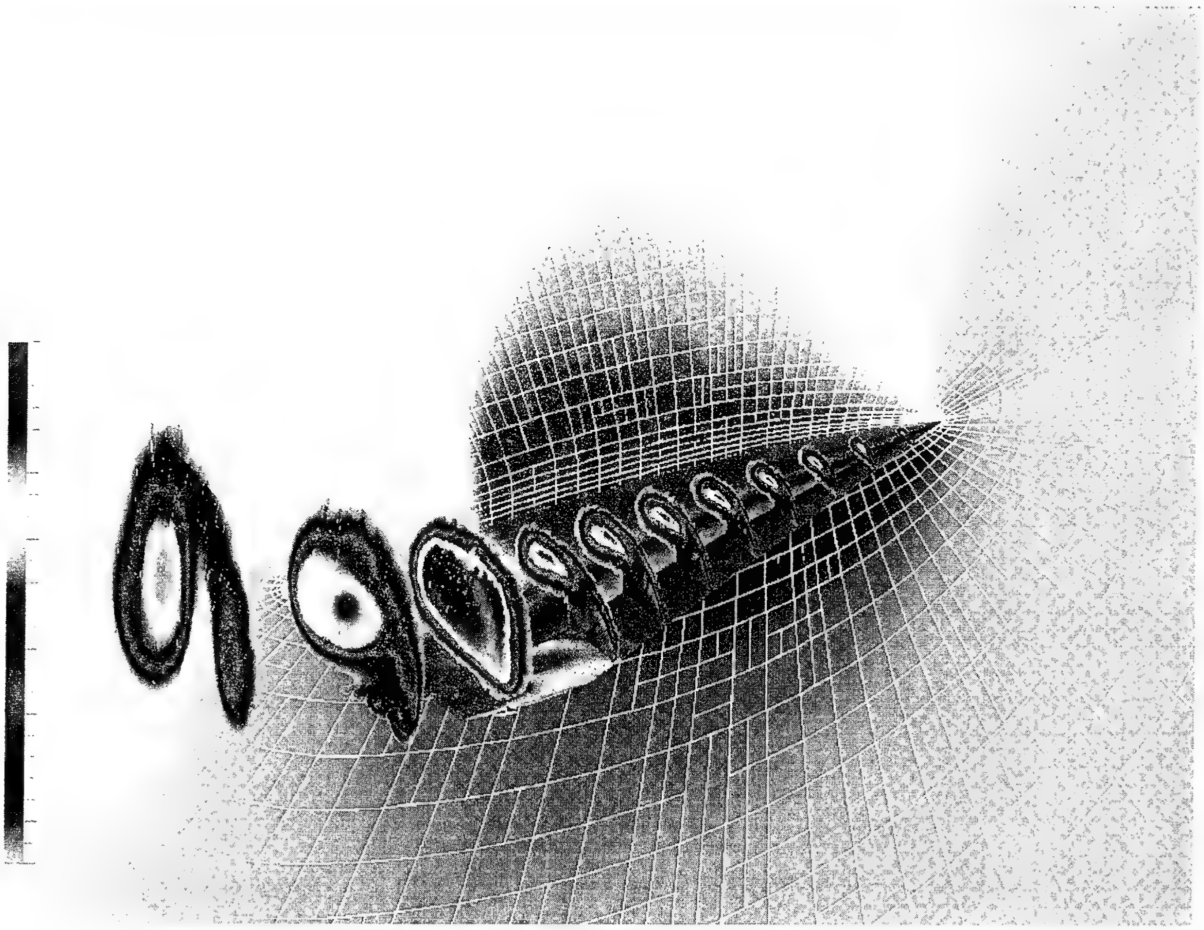


Figure 4. *Vortex structure on leeward side of delta wing, visualized as total pressure loss. ($M_\infty = 0.85, \alpha = 20°$)*

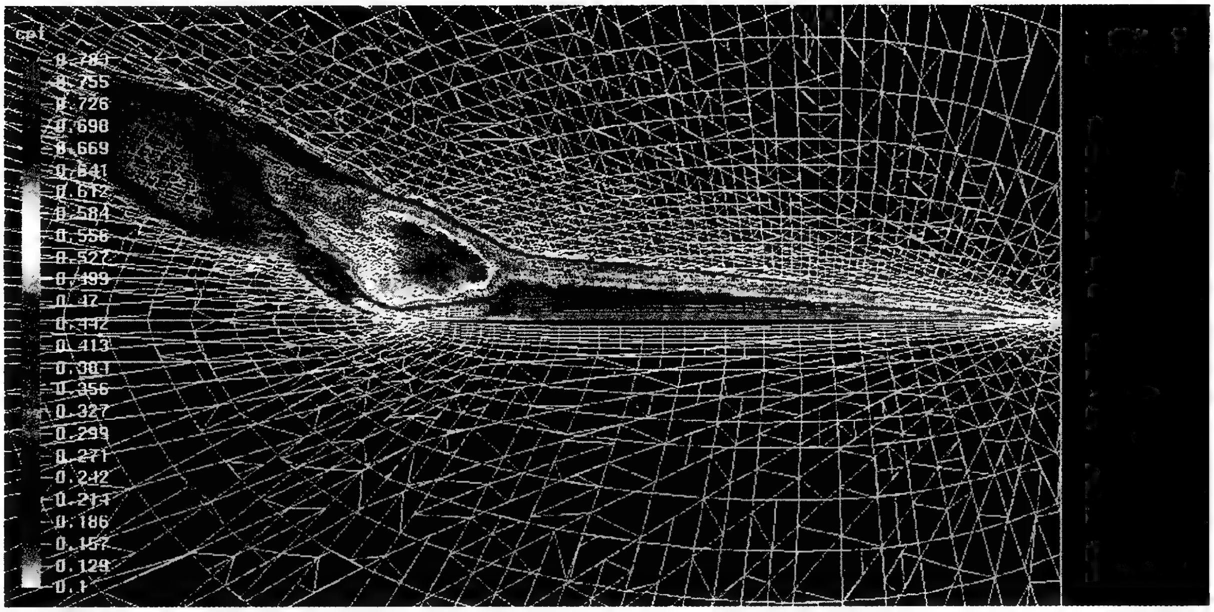


Figure 5. *Total pressure loss and adapted grid in cross-section through primary vortex core.* $(M_\infty = 0.85, \alpha = 20°)$

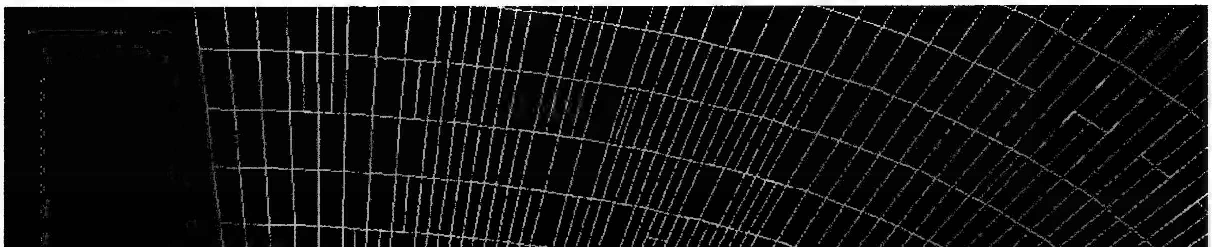


Figure 6. *Total pressure loss and adapted grid in cross-section at 70% chord.* $(M_\infty = 0.85, \alpha = 20°)$

Parallel algorithms for DNS of compressible flow

Martin Streng, Hans Kuerten, Jan Broeze and Bernard Geurts
Department of Applied Mathematics, University of Twente
P.O.Box 217, 7500 AE Enschede, The Netherlands

Abstract

We indicate that the use of higher order accurate spatial discretization is necessary to obtain sufficiently accurate DNS for the validation of subgrid models in LES. Furthermore, we pay attention to the efficiency of the implementation of these discretizations on several parallel platforms. In order to illustrate this, we consider compressible flow over a flat plate. We give *a priori* test results for LES of this flow.

1 Introduction

One of the most challenging problems in Computational Fluid Dynamics (CFD) is the accurate and efficient simulation of turbulent flows for relevant industrial applications. The behaviour of these flows is governed by the Navier-Stokes equations. However, because these applications usually involve complex geometries and flow-fields, the computational resources required for directly solving the Navier-Stokes equations are far beyond the resources which will be available in the foreseeable future. In this paper we will focus on turbulent compressible flow-problems in simple geometries. In order to tackle these problems with presently available computers, three different aspects must be considered: the modelling of turbulent flows, the numerical methods used to perform calculations with these models, and the implementation of these methods on suitable computer platforms.

As remarked above, direct solution of the Navier-Stokes equations (DNS) is impossible for relevant industrial applications, due to the high computational requirements. Therefore, one might use instead the Reynolds averaged Navier-Stokes (RaNS) equations in which only the statistically stationary flow is calculated and the effects of turbulence are modelled by a so-called turbulence model. However, this leads in general to quite inaccurate results since the presently available turbulence models are inadequate for more complicated flow phenomena like shock-boundary layer interaction and massive separation. A solution to this problem could be provided by Large Eddy Simulation (LES). In LES only the large eddies are calculated, while the effects of the smaller eddies, which are thought to be universal and not geometry-dependent, are described by a subgrid model.

However, before LES can be used as a tool in flow simulation, the subgrid model has to be systematically validated. This validation is usually carried out by comparing LES results with filtered DNS results for simple geometries and fairly low Reynolds numbers. In Section 2 we present *a priori* test results for LES of compressible flow over a flat plate for various subgrid models, including eddy-viscosity models, the similarity model and dynamic models. In the future also *a posteriori* tests will be carried out fot this flow, as has been done e.g. by Vreman et al. [1] for the compressible mixing layer.

The numerical methods to perform the DNS are discussed in Section 3. The a priori test results are based on DNS performed using a second-order finite volume spatial discretization. It is indicated that the use of higher order spatial discretizations makes it possible to obtain more accurate DNS results. However, the use of higher order central differencing discretizations, without numerical dissipation, is not without trouble. Besides the occurrence of stability problems, higher order discretizations lead to wide stencils, which, in combination with a domain-decomposition strategy, seriously affects the parallel efficiency of the resulting algorithm.

In Section 4 the parallel efficiency will be illustrated using some implementations of the DNS solver on various parallel platforms, including distributed as well as shared memory systems, and a mixture of these types. Since many parallel platforms use cache-based processors, we consider some

*Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms"
held in Seville, Spain, from 2-5 October 1995, and published in CP-578.*

aspects of implementation of the flow-solver on these processors. We show that careful use of cache in the implementation of our type of discretizations can lead to considerable performance gain.

2 Modelling of turbulent flow

The equations describing compressible flow are the well known Navier-Stokes equations, which represent conservation of mass, momentum and energy:

$$\begin{aligned} \partial_t\rho \; &+ \; \partial_j(\rho u_j) = 0 \\ \partial_t(\rho u_i) \; &+ \; \partial_j(\rho u_i u_j) + \partial_i p - \partial_j \tau_{ij} = 0 \\ \partial_t e \; &+ \; \partial_j((e+p)u_j) - \partial_j(\tau_{ij}u_i - q_j) = 0 \end{aligned} \qquad (1)$$

Here the symbols ∂_t and ∂_j are abbreviations of the partial differential operators $\partial/\partial t$ and $\partial/\partial x_j$ respectively. The components of the velocity vector are denoted by u_i, while ρ is the density and p the pressure which is related to the total energy density e by:

$$p = (\gamma - 1)\{e - \frac{1}{2}\rho u_i u_i\} \qquad (2)$$

in which γ denotes the adiabatic gas constant. The viscous stress tensor τ_{ij} is a function of temperature T and velocity vector $\mathbf{u}$

$$\tau_{ij}(T,\mathbf{u}) = \frac{\mu(T)}{Re}(\partial_j u_i + \partial_i u_j - \frac{2}{3}\delta_{ij}\partial_k u_k) \qquad (3)$$

where $\mu(T)$ is the dynamic viscosity for which we either use Sutherland's law for air or treat it as a constant. In addition q_j represents the viscous heat flux vector, given by

$$q_j(T) = -\frac{\mu(T)}{(\gamma-1)RePrM^2}\partial_j T \qquad (4)$$

where Pr is the Prandtl number. Finally, the temperature T is related to the density and the pressure by the ideal gas law

$$T = \gamma M^2 \frac{p}{\rho} \qquad (5)$$

These governing equations have been made dimensionless by introducing a reference length L_0, velocity u_0, density ρ_0, temperature T_0 and viscosity μ_0. The values of the Reynolds number $Re = (\rho_0 u_0 L_0)/\mu_0$ and the Mach number $M = u_0/a_0$, where a_0 is a reference value for the speed of sound, are given separately.

A Direct Numerical Simulation (DNS) is based on a discretisation of (1) whereas the governing equations for large eddy simulation (LES) are obtained by applying a spatial filter to these equations. A filter operation extracts the large scale part $\bar{f}$ from a quantity f:

$$\bar{f}(\mathbf{x}) = \int_\Omega G_\Delta(\mathbf{x},\xi) f(\xi) d\xi \qquad (6)$$

where Ω is the flow domain and Δ denotes the filter width of the kernel G which is assumed to be normalized, i.e. the integral of G over Ω equals 1 independent of $\mathbf{x}$. For compressible flow Favre [2] introduced a related filter operation $\tilde{f} = \overline{\rho f}/\bar{\rho}$.

The filtered Navier-Stokes equations contain so-called subgrid-terms, which cannot be expressed in the filtered flow variables, and have to be modelled with subgrid-models. In this paper we will mainly focus on the modelling of the subgrid-terms in the momentum equations, which can be expressed in the turbulent stress tensor, defined as

$$\bar{\rho}\tau_{ij} = \overline{\rho u_i u_j} - \overline{\rho u_i}\,\overline{\rho u_j}/\bar{\rho} = \bar{\rho}(\widetilde{u_i u_j} - \tilde{u}_i\tilde{u}_j), \qquad (7)$$

where $\tilde{\mathbf{u}}$ is the filtered velocity vector. This turbulent stress tensor has several algebraic properties which can be used in the construction and qualification of subgrid-models [3, 4]. Expressions for the subgrid-terms in the energy equation can be found in ref. [5]. They can be neglected in simulations at low Mach numbers, but have to be modelled at high Mach numbers.

In total six models for the turbulent stress tensor τ_{ij} as it appears in the subgrid-terms in the momentum equations will be investigated and compared in this paper. The first subgrid-model is the Smagorinsky model

$$\bar{\rho}\tau_{ij}^{(1)} = -\bar{\rho}C_S^2\Delta^2|\tilde{S}|\tilde{S}_{ij}, \qquad (8)$$

where $\tilde{S}^2 = \frac{1}{2}\tilde{S}_{ij}^2$ with $\tilde{S}_{ij}$ the compressible strain rate, based on the Favre-filtered velocity. C_S is the Smagorinsky constant, which we choose equal to 0.17 as suggested in literature. Δ denotes the filter width, which separates the resolved and subgrid-scales. The major short-coming of the Smagorinsky model is its excessive dissipation in regions where the flow is laminar [6]. The similarity model, formulated by Bardina *et al.* [7], is based on a similarity assumption. Application of the definition of $\bar{\rho}\tau_{ij}$ to the filtered variables $\bar{\rho}$ and $\overline{\rho u_i}$ yields the similarity model [7]:

$$\bar{\rho}\tau_{ij}^{(2)} = \overline{\bar{\rho}\tilde{u}_i\tilde{u}_j} - \overline{\bar{\rho}\tilde{u}_i}\,\overline{\bar{\rho}\tilde{u}_j}/\bar{\bar{\rho}}. \qquad (9)$$

The gradient model is derived with use of Taylor expansions of the filtered velocity [8]. The lowest order term in Δ in this expansion can be proposed as subgrid-model:

$$\overline{\rho}\tau_{ij}^{(3)} = \frac{1}{12}\overline{\rho}\Delta^2(\nabla\tilde{u}_i)(\nabla\tilde{u}_j). \qquad (10)$$

The similarity and gradient model correlate much better with the turbulent stress tensor than the Smagorinsky model (see [9] and section 2.1). However, while the Smagorinsky model is too dissipative in transitional regions, the similarity and gradient model are not sufficiently dissipative in turbulent regions.

The dynamic procedure overcomes the excessive dissipation of the Smagorinsky model and adds sufficient dissipation to the similarity and gradient models. We consider three dynamic models. The dynamic eddy-viscosity model [3] is obtained when the model constant C_S in the Smagorinsky model is replaced by a coefficient which is dynamically obtained and depends on the local structure of the flow. In order to calculate the dynamic coefficient $\tau_{ij}^{(1)}$ is substituted in the Germano identity, which is a relation between the turbulent stress tensor for different filter widths [3]. The second dynamic model is the dynamic mixed model, in which a relatively accurate representation of the turbulent stress by the similarity model and a proper dissipation provided by the dynamic eddy-viscosity concept are combined [10]. The dynamic model coefficient is obtained by substitution of the base mixed model, $\tau_{ij}^{(2)} + \tau_{ij}^{(1)}$, in the Germano identity. Another dynamic model is the dynamic Clark model [11]. In this case the base model is the Clark model, $\tau_{ij}^{(3)} + \tau_{ij}^{(1)}$, and the model coefficient C_S is obtained by substitution of this model in the Germano identity.

2.1 Results

We consider flat plat flow at $Re = 1000$ based on the initial displacement thickness δ_* and the other reference scales are equal to the initial far-field values. We choose $M = 0.5$ and consider a temporal simulation in a cubic domain of size 30. A forcing term corresponding to the compressible similarity solution of the boundary layer equations is added. The mean initial field also equals this similarity solution, to which the dominant 2D mode and a pair of equal and oblique 3D modes are added with amplitude 10^{-3} and amplitude-ratios $(1/2, 1/4, 1/4)$ respectively. For validation purposes the linear growth rates of the instabilities were recovered with a relative error well within 1 percent on a grid with 128^3 cells, uniform in the stream- and spanwise directions and clustered near the isothermal, no-slip wall in the normal direction. A second order accurate finite volume method was used.

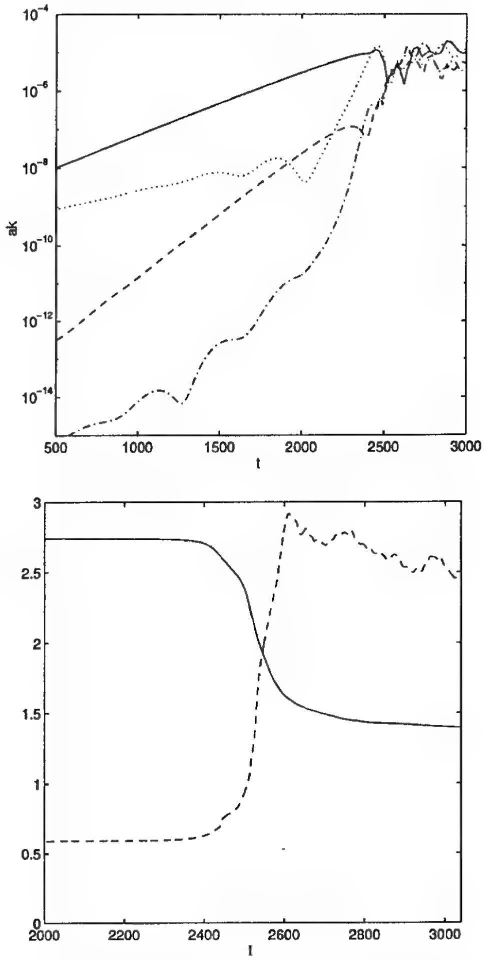


Figure 1: Modes of kinetic energy (a) [(1,0):solid, (2,0): dashed, (1,1): dotted, (2,2): dash-dotted] and shape-factor (solid), skin-friction (dashed) versus time t (b)

Results from a DNS on 128^3 cells are shown in Figure 1. The persisting symmetry in the spanwise direction was exploited in order to reduce the computational effort. The evolution of the amplitude of some modes of the kinetic energy (Fig. 1) clearly displays the initial linear regime with an exponential growth of the instabilities. The corresponding large-scale structures which emerge subsequently interact in the nonlinear regime and give rise to a rapid transition in which many modes become simultaneously important. A broad spectrum is generated and a developed turbulent flow results in which the individual modes display an erratic time-dependence. To represent this scenario in a different way, the shape-factor and the skin-friction are shown in Figure 1. The resolution is adequate in the linear and transitional stages with a fall off of 10 decades or more in the spectrum of the kinetic energy. However, at the onset of turbulent flow and in the developed stages a fall off of no more than 6-7 decades was observed. Hence, the results in the turbulent regime are expected to be only qualitatively correct and further grid refinement is needed.

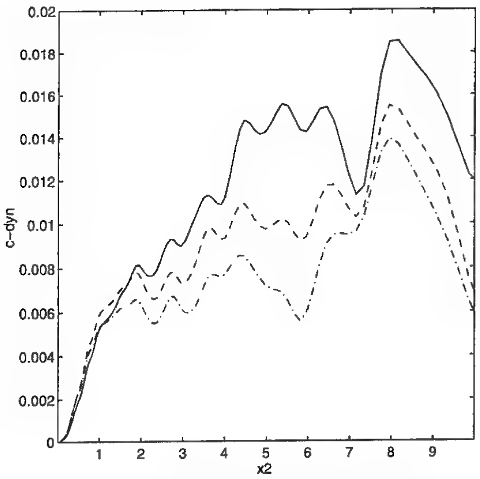


Figure 2: Dynamic coefficients : Germano (solid), dynamic mixed (dashed) and dynamic Clark (dash-dotted).

In order to obtain a first impression of the quality of the various subgrid models for this flow we focus on the correlation between $\bar{\rho}\tau_{12}$ and the corresponding modelled component of the turbulent stress tensor. We use a filter-width equal to four grid-cells and a special filtering near the wall which prevents the filter to extend inside the wall. The models are tested both in the transitional and in the turbulent regime. The similarity- and gradient model as well as the dynamic mixed and dynamic Clark model show a high correlation of about 0.9. The Smagorinsky and dynamic eddy-viscosity models show a poor correlation of about 0.3. The eddy-viscosity contribution in the dynamic mixed and dynamic Clark model does not destroy the high correlation. In Figure 2 we compare the dynamic coefficients for the three dynamic models at $t = 2700$. The coefficients are averaged over the homogeneous directions. We observe that the Germano coefficient is larger than the coefficient associated with the other two dynamic models. Moreover, all coefficients drop to zero in the near-wall region which is appropriate for wall-bounded shear layers.

3 Numerical method

As has been remarked in the previous section, the DNS results in the turbulent regime are expected to be only qualitatively correct, and further grid refinement is needed. However, the number of grid-cells used is already fairly large for presently available computer resources. Instead of refinement, we presently consider the use of higher order discretization methods. The aim is to obtain a more accurate DNS with a moderate number of points. However, this is not without problems. One drawback is that high order methods lead to wide stencils, which decreases the parallel efficiency of the resulting code, as we will see in the next section. Another problem associated with these methods is that the discretisation of the convective and the viscous flux must be carefully constructed in order to avoid instabilities. This is especially present in central differencing methods, and is not only related to the occurrence of π-modes, but also to adequate damping of aliasing errors.

3.1 Spatial discretization

Consider an orthogonal grid with points $x_{i,j,k}$, which is uniform in x and z direction. We use the following central differencing discretization of the $\frac{\partial}{\partial x}$-operator:

$$(\frac{\partial f}{\partial x})_{ijk} = \sum_{n=-d}^{d} w_{j,n}^{diff} a_{i+n,j,k}, \qquad (11)$$

where

$$a_{ijk} = \sum_{n,m=-d}^{d} w_{j,n,m}^{av} f_{i,j+n,k+m}. \qquad (12)$$

Here the weights w^{diff} are derivative weights, and w^{av} are average weights. Due to the uniformity in x and z direction they only depend on j. The quantities a represent the average of the function f over a stencil in $j-k$ direction. For the convective flux we use a stencil with N_c points, and the weights w^{av} are constructed such that π modes in the j and k direction are filtered out, and moreover that polynomials up to degree $N_c - 1$ are invariant under the averaging. The derivative weights w^{diff} are such that polynomials up to degree N_c are exactly differentiated. The resulting discretization has order N_c on uniform grids. The π-modes in i-direction are damped by the viscous derivative. The viscous flux is discretized using repeated differentiation. The inner derivative is calculated on a staggered grid. Both the inner and the outer derivatives are discretized analogously as in the convective flux, on N_v points, except that now π-modes are not filtered out. Both derivatives are then of order $N_v - 1$, but due to symmetry, on a uniform grid, the viscous flux is discretized up to order N_v.

Due to the nonlinearity in the convective flux, high frequency modes arise from a low-frequency initial state. In physical reality, these are damped by the viscous effects in the fluid. In the numerical simulation, however, two difficulties arise. The first is that both the convective and the viscous flux are calculated inaccurately. In our central differencing discretisations, on relatively coarse grids, a situation may arise in which the numerical viscous terms do not have enough dissipation to damp the numerical convective terms, giving rise to instabilities. The second difficulty is that due to the finite grid-spacing, there is a maximum wavenumber which can be represented on the grid. Modes with a higher wavenumber appear as low-frequency modes on the grid. Therefore, numerically, the effective energy contained in the low-frequency modes can be increased during the onset of turbulence. One remedy could be to take a grid that is sufficiently fine to represent the highest mode which due to physics would emerge in the simulation. Another possibility is to use upwind-biased discretizations of the convective flux, as has been done by Rai and Moin [12]. We have used a discretisation of the viscous flux with a wider stencil than necessary to achieve the desired order of accuracy. In this way we constructed a better approximation of the viscous flux. As an example, we were able to calculate a full transition to turbulence on 96^3 points using a fourth order method on a 5^3-points stencil for the convective flux, and repeated application of a fourth order method on 6^3 points for the viscous flux, resulting in an 11^3-points stencil, whereas repeated application of a 4^3 points operator for the viscous flux on this grid failed. At this moment, further investigation is needed to understand this phenomenon more clearly.

The DNS mentioned in the previous section has been calculated at Mach number 0.5. In the future we intend to perform DNS at higher Mach numbers. For that purpose we need to be able to capture shocks. This can be done by switching to upwind discretizations in the presence of a shock, which has been applied succesfully to the supersonic compressible mixing layer, cf. ref. [13]. In that application a fourth order central difference operator has been used for the convective term, which was replaced by a third order accurate upwind scheme in the presence of a shock. See Figure 3. In this way it is possible to capture time-dependent shocks which appear spontaneously after the transition to turbulence.

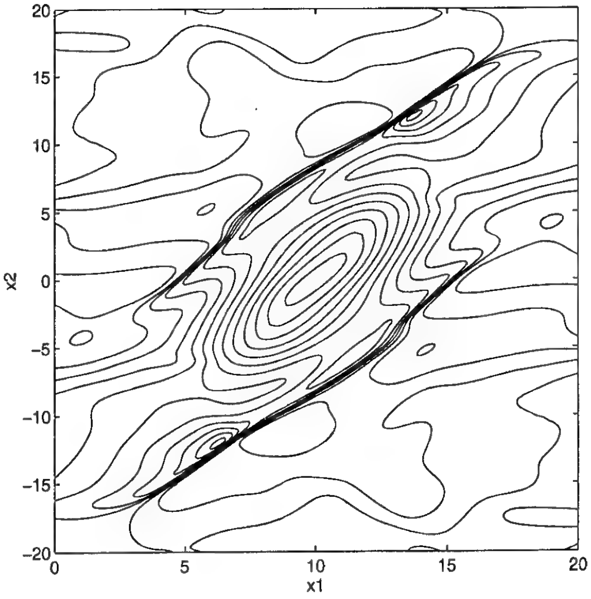


Figure 3: Shock-capturing in 3D turbulent mixing-layer.

3.2 Time integration

For the time integration of the resulting discretized equations we use an explicit 4-stage Runge Kutta method. We also studied the use of a second-order accurate implicit method. The system of equations resulting from the implicit discretization is solved by means of pseudo-time stepping and accelerated by local pseudo-time stepping and a nonlinear multigrid technique. Since we use central spatial discretizations and no artificial dissipation is added to the equations, the smoothing method is less effective than in the traditional use of multigrid in steady-state calculations. In the laminar regime and in the first stages of turbulence the implicit method provides a speed-up of a factor of 2 relative to the explicit method on a relatively coarse grid (64^3). At increased resolution this speed-up is enhanced correspondingly. See [14].

4 Parallel implementation of the explicit solver

In this section we consider some implementational aspects of the explicit solver. We use a simple domain-decomposition technique to obtain an implementation on a parallel computer. This is explained in the first subsection. In the next subsection we discuss how the parallel efficiency of the resulting code depends on the spatial discretization. We distinguish between the *intrinsic* efficiency of an algorithm, and the *hardware* efficiency. The former is related to the algorithm only, whereas the latter tells us how good a certain algorithm performs on certain hardware. The quantity which is usually called the efficiency is the product of these efficiencies. We show that the intrinsic efficiency of the algorithm decreases as the order of the spatial discretization increases. We illustrate these concepts by some performance results obtained from implementations on 3 different parallel machines, viz. the Cray T3d, the Intel Paragon and the SGI Power Challenge array. Closely related to the concept of efficiency is the scalability. We discuss the scalability in the sense of Amdahl and Gustafsson (see e.g. ref. [15]).

4.1 Domain decomposition

Suppose our computational domain consists of $N_x \times N_y \times N_z$ gridpoints. This domain is divided into $B_x \times B_y \times B_z$ blocks. For a distributed memory computer, we assume that each block is allocated on a separate processor. If the total size of the stencil used for the discretisation is $(2d+1)^3$ (recall that we use central differences, cf. (11),(12)), then a point which has a distance less than $d+1$ grid-points from the boundary of a block not coinciding with the boundary of the physical domain, is called an interior boundary point. This definition can easily be extended to other discretisation methods. For the computation of the fluxes for the interior boundary points, some values of the flow-quantities which reside on processors dealing with neighbouring blocks are needed. To store these quantities, each block is dressed with d dummy-layers. In order to retain the second-order accuracy of the time-integration method, at each stage in the Runge-Kutta time-integration, these dummy-layers have to be transferred between the various processors. It may be clear that the amount of communication increases with the size of the stencil.

Not only the amount of communication is affected by the size of the stencil, but also the number of floating point operations increases with increasing stencil-size. To see why, recall the general form of the $\frac{\partial}{\partial x}$-operator, eq. (11)–(12). This derivative is computed as a one-dimensional derivative acting on two-dimensional averages over y and z. For the derivative in an internal boundary point these averages have to be computed for points in the dummy-layers as well. But these averages are also computed by the processors dealing with the neighbouring block in order to contribute to the $\frac{\partial}{\partial x}$ derivative of some points in that block. For a discretization on a stencil with $N_x \times N_y \times N_z$ points, careful counting reveals that the number of floating-point operations for the computation of one derivative is

$$(3N_xN_yN_z + 4dN_yN_z + 2dN_xN_z + 4d^2N_z)(2d-1).$$

Note that this expression is not symmetric in N_x, N_y, N_z. For the other derivatives the discrete averaging and differentiation operators can be applied in such an order that the same expression is valid. In the case $N_x = N_y = N_z = N$, this reduces to

$$(3N^3 + 6N^2d + 4d^2N)(2d-1). \qquad (13)$$

Now consider e.g. a given partition of the computational domain into B^3 equal blocks, each containing $(N/B)^3$ points. Then the total number of floating-point operations to compute a $\frac{\partial}{\partial x}$ for all grid-points is

$$3((\frac{N}{B})^3 + 6(\frac{N}{B})^2d + 4d^2\frac{N}{B})(2d-1)B^3,$$

which is obviously greater than (13).

4.2 Parallel efficiency

To quantify the considerations of the previous paragraph, we define the concept of intrinsic efficiency. Consider a given partition of the computational domain into $B_x \times B_y \times B_z$ blocks. Denote the total number of floating point operations for a given number of timesteps by $f(B_x, B_y, B_z)$. Then the intrinsic efficiency σ_{intr} is given by

$$\sigma_{\text{intr}} = \frac{f(1,1,1)}{f(B_x,B_y,B_z)}. \qquad (14)$$

Note that, on a shared memory machine, if we use fine-grained parallellism (on do-loop level), we could define $\sigma_{\text{intr}} = 1$.

We can estimate the dependence of the intrinsic efficiency on the size of the stencil just by counting the number of floating-point operations for various block-sizes (by using expressions like (13)). In Figure 4 this has been done for several central differencing discretizations, using equal shapes and sizes for all blocks. From the pictures it can be seen that the efficiency decreases rapidly if the stencil-size grows. Due to the wider stencil, application of higher-order discretizations results in more floating-point operations, but this performance penalty is even more severe on distributed memory systems, where also a decrease of parallel performance occurs. As an example, consider a central differencing second order $\frac{\partial}{\partial x}$ operator on a 3-point stencil as compared to a central differencing fourth order $\frac{\partial}{\partial x}$ operator on a 5-point stencil. To compute the former derivative on a single-cpu machine costs approximately $5/9 \approx 0.56$ times of the time to compute the latter, whereas on e.g. a $64 \times 64 \times 32$ grid and 128 processors on a distributed memory machine this ratio is approximately 0.33.

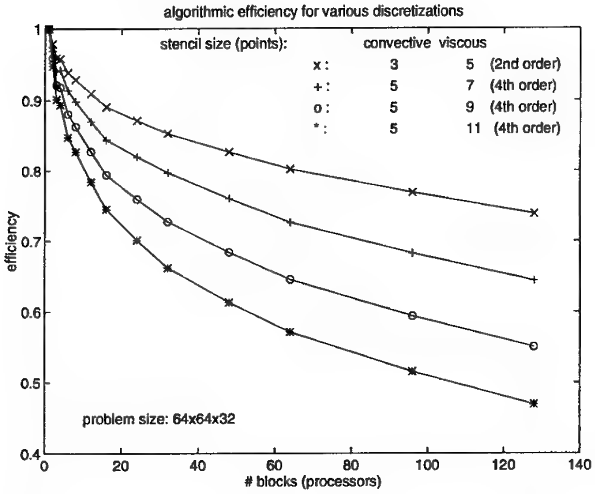


Figure 4: Intrinsic efficiency for various spatial discretizations

The intrinsic efficiency deals with the parallelizability of a given algorithm, regardless of any machine. In fact it gives the maximum speed-up that can be achieved for the algorithm. In a real implementation the speed-up will be less, due to e.g. the finite bandwidth of the machine. To quantify this, we now define the hardware efficiency σ_{hw}. Suppose the CPU time to perform a certain number of timesteps on one processor using one block is $T(1,1,1)$. Then, using $B_x B_y B_z$ processors, the CPU time cannot be shorter than

$$\frac{T(1,1,1)}{B_x B_y B_z \sigma_{\text{intr}}(B_x, B_y, B_z)}.$$

In general, due to the finite communications bandwidth of the machine, the simulation will last longer, say $T(B_x, B_y, B_z)$ seconds. Then the hardware-efficiency σ_{hw} is

$$\sigma_{\text{hw}} = \frac{T(1,1,1)}{T(B_x, B_y, B_z) B_x B_y B_z \sigma_{\text{intr}}(B_x, B_y, B_z)}. \tag{15}$$

The traditional (total) efficiency σ is the product

$$\sigma = \sigma_{\text{hw}} \sigma_{\text{intr}}. \tag{16}$$

Note that, in general, these efficiencies not only depend on the number of blocks in each direction, but also on the number of points per block in each direction, i.e. on the actual shape of the blocks. This is not only due to the ratio of interior boundary points as compared to the interior points of each block, but also because many processors perform better on long inner loops in the code, due to vectorisation or pipelining.

The efficiency σ is related to scalability in the sense of Amdahl, meaning that a problem which is solved on one processor in T_1 seconds is solved on P processors in $T_1/P\sigma$ seconds. We define one notion of efficiency related to scalability in the sense of Gustafson. Suppose we solve a problem with N gridpoints on one processor in T_1 seconds, and a problem with PN gridpoints in T_P seconds. Then the efficiency σ_G is

$$\sigma_G = \frac{T_P}{T_1}. \tag{17}$$

These concepts are illustrated in Figure 5. Here we performed 5 timesteps on a $64 \times 64 \times 32$ grid, with a 5 point central differencing discretization of the convective flux, and a repeated application of a four-point central differencing for the viscous flux, resulting in a total stencil containing $7 \times 7 \times 7$ points. Plotted are the intrinsic efficiency and the total efficiency. Because it was not possible to execute the program on 1 or 2 CPUs on the Paragon, the efficiencies are based on the timings for the 4-processor run. We used 2 different distributed memory machines, viz. the Cray T3d and the Intel Paragon. On these machines, explicit message-passing has been employed. The actual CPU-times for the runs are tabulated in Table 1. A dash indicates that it had not been possible to perform the run on the indicated number of processors, either because the processors do not have enough memory (in the case of 1 and 2 processors on the Paragon) or because the indicated number of processors was not available on that machine. The CPU times are depen-

# proc.	T3d	Paragon
1	207.5	–
2	109.6	–
4	58.3	90.6
6	42.7	65.2
8	33.4	50.3
12	23.8	36.8
16	17.1	27.4
24	13.5	20.4
32	9.9	15.7
48	7.5	11.7
64	6.0	11.1
96	4.8	7.6
128	3.8	–

Table 1: CPU times in seconds (averaged over several block-divisions).

dent on the actual shape of the blocks. Therefore in Table 1 we averaged over some block-divisions which give roughly the same (approximately best) CPU-time. This dependency is illustrated in Table 2 for the case of 8 blocks. All timings are accurate to about 5 %. It can be seen that subdivisions with an equal number of blocks in all directions are optimal. In general, better subdivisions are obtained by using fewer blocks in x-direction. This is partly due to the algorithm, since an asymmetry is introduced by the sequence of averaging-operators in the derivative-calculations, and partly due to software-pipelining in the processors, which is reflected in the megaflop-rates (between parentheses).

$B_x \times B_y \times B_z$	T3d	Paragon	
$1 \times 1 \times 8$	34.9 (77)	58.0 (47)	335
$1 \times 8 \times 1$	34.3 (82)	54.4 (52)	353
$8 \times 1 \times 8$	39.2 (77)	67.3 (45)	378
$1 \times 2 \times 4$	32.4 (80)	50.9 (52)	323
$1 \times 4 \times 2$	31.6 (83)	50.0 (53)	328
$2 \times 1 \times 4$	33.0 (79)	52.9 (50)	327
$2 \times 4 \times 1$	31.7 (84)	51.5 (53)	335
$4 \times 2 \times 1$	32.9 (80)	54.6 (50)	327
$4 \times 1 \times 2$	32.9 (82)	55.4 (49)	339
$2 \times 2 \times 2$	31.1 (84)	50.3 (53)	325

Table 2: CPU times for various subdivisions into 8 blocks. Between parentheses the Mflop-rates. The last column is the number of millions of floating point-operations to be performed for each block.

From the pictures it can be seen that on the T3d and the Paragon, the machine efficiency is somewhat lower than the algorithmic efficiency. This means that increasing the algorithmic efficiency by e.g. exchanging information between the processors after every calculation of averages will not result in a substantially faster execution of the code. Further, all efficiencies eventually approach zero as the number of processors approaches infinity. It can be shown (using expressions like (13)) that the intrinsic efficiency drops as $B^{-2/3}$, where B is the total number of blocks. However, σ_G remains nearly constant, as is shown in Table 3. Here each block contains 32 × 16 × 16 points. From this table it follows that, using this algorithm, doubling the size of the problem and the number of processors results in equal computation times. This can also be shown if in (17) the times T_P and T_1 are calculated as ideal, i.e. assuming no communications delays. Then $\sigma_G = 1$.

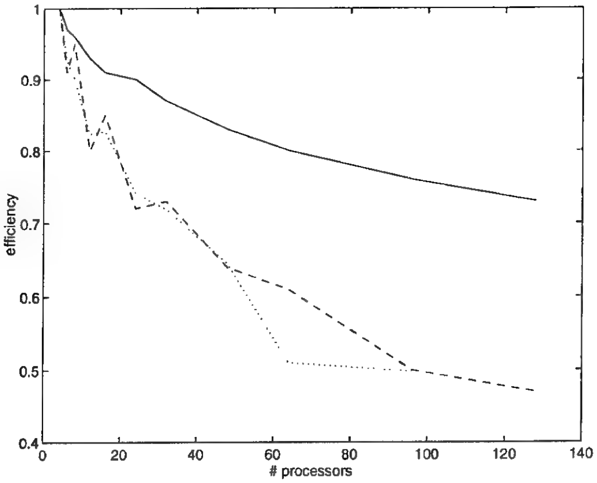


Figure 5: Efficiency for the T3d (dashed) and the Paragon (dotted). The solid line is the intrinsic efficiency.

$B_x \times B_y \times B_z$	T3d	Paragon
$1 \times 2 \times 1$	16.2 (21.1)	26.9 (12.7)
$1 \times 4 \times 1$	16.3 (42.1)	27.0 (25.4)
$1 \times 4 \times 2$	16.4 (83.6)	27.3 (50.2)
$2 \times 4 \times 2$	16.5 (166)	27.6 (99.4)
$2 \times 8 \times 2$	16.5 (332)	27.4 (200)
$2 \times 8 \times 4$	16.6 (661)	27.7 (396)
$2 \times 8 \times 6$	16.6 (991)	27.8 (592)
$4 \times 8 \times 4$	16.6 (1322)	–

Table 3: CPU times and Megaflop-rates (between parentheses) for increasing domain-sizes illustrating that σ_G remains approximately constant.

From the above results it can be concluded that the T3d and the Paragon show comparable efficiencies for this algorithm, the T3d being about 40 % faster.

Besides the implementation on the T3d and the Paragon, we have made a preliminary implementation on the SGI Power Challenge Array. This machine consists of 4 nodes each comprised of a 16-CPU shared memory parallel machine. We used explicit message-passing between the nodes. On each node, fine-grained parallelism has been employed using the vendor-supplied parallelizing compiler. The combination of fine-grained parallelism and explicit message passing is not entirely trivial. On the one hand, using fine-grained parallelism results in an algorithmic efficiency of 1, since no additional floating-point operations are introduced. Therefore, this form of parallelism seems to be promising at first sight. On the other hand, however, parallelizing a do-loop containing only a few iterations (in the order of magnitude of the number of grid-points in one directions) causes much system-overhead, and seriously affects pipelining efficiency. Moreover, suboptimal speedup can arise due to the cache-coherency mechanism. The use of explicit message-passing has two disadvantages, namely an algorithmic efficiency less than one, and usually a slow data-transfer. The advantage of explicit message-passing as compared to fine-grained parallelism is that parallelization takes place on a (much) higher level, leading to less system overhead.

As an example, consider a problem with $64 \times 64 \times 32$ grid-points (the same as discussed above). With 4 processors on one node working on one block, this yields an execution time of 23 seconds for 5 Runge-Kutta timesteps, whereas on 4 nodes with 4 blocks $(1 \times 2 \times 2)$ and one processor per node the execution time is 18 seconds. As another example, we compare the subdivision into $1 \times 2 \times 2$ and $2 \times 4 \times 2$ blocks, both running on 4 nodes. In the first case, each node deals with 1 block, and in the second case each node does the computations for 4 blocks, and uses 2 processors for each block. So in that case the distributed memory model is adopted also *within* each single node. It appears that the latter case has a shorter execution time. It may be clear that some restructuring of the code is necessary in order to obtain reasonable performance. This will be the subject of another paper [16].

4.3 Optimization for cache-machines

In many parallel machines the processors use a hierarchical memory structure, consisting of a small amount of memory with a short access time (the cache) and a large amount of main memory with much longer access time. This long access time is the main reason why the performance of these machines is way below their (often impressive) peak. In the implementation of a numerical algorithm, it is essential to use the cache efficiently. Therefore, the number of load and store operations should be kept to a minimum, and quantities which are loaded from main memory should be reused as much as possible before being restored. Further, since elements from main memory are loaded into cache in chunks of a few consecutive elements, do-loops should be arranged such that main memory is traversed linearly (as is also necessary for efficient use of traditional vector-processors). Moreover, it will enable software-pipelining on RISC-processors, resulting in substantially faster execution.

To illustrate this, we compare two different ways to calculate the viscous flux. In the first method (method A) the various derivatives of the velocity fields and the temperature are calculated *consecutively*, and the viscous stress tensor and viscous heat flux are assembled and stored. Then the outer derivatives of the viscous flux are calculated, again consecutively. The resulting code is very well vectorizable and consists of very simple do-loops. In the second method (method B), we use the following observation. In the calculation of the derivatives, some averages can be used to contribute to various derivatives. Moreover, for all derivatives, the averaging weights in one direction are equal. Therefore we calculate all inner derivatives simultaneously, which also has the advantage that e.g. a vector u_1 needs to be loaded only once for the calculation of all its derivatives. An analogous fact holds for the weights. Further, the derivatives are not stored, but directly used to assemble the stress tensor and the heat flux. After that, all outer derivatives are calculated simultaneously. This results in about 30% less floating point operations, and substantially less load and store operations, resulting in better memory-performance. The drawback is the occurrence of (much) more complicated do-loop bodies, which puts a severe demand on the compiler in order to obtain suitable pipelining. It appears that on the T3d and the Paragon there is hardly any performance gain, and the performance is only about 20 % of peak. On one R8000 processor in the SGI Power Challenge (coupled to 4 MBytes of cache), the CPU-time of method B is half that of method A, with a performance of about 37 % of peak (110 Mflops). More details are to be found in ref. [17].

Acknowledgement

The time for the computations on the T3d was provided by the Stichting Nationale Computerfaciliteiten (National Computing Facilities Foundation, NCF), which is financially supported by the Nederlandse Organisatie van Wetenschappelijk Onderzoek (Netherlands Organization for Scientific Research, NWO). One of the authors (HK) thanks the Institute for Fluid Dynamics at ETH Zürich for its hospitality during his stay there. The use of the Paragon has been made possible by courtesy of Prof. B. Müller, ETH Zürich. By courtesy of Silicon Graphics Inc. we were able to use the European Power Challenge Array of the SGI SuperComputer Technologies Centre in Cortaillod. We would like to thank Ruud van der Pas (SGI) for his assistance.

References

[1] B. Vreman, B. Geurts, H. Kuerten, J. Broeze, B. Wasistho and M. Streng, "Dynamic subgrid-scale models for LES of transitional and turbulent compressible flow in 3-D shear layers," Turbulent Shear Flow, (1995)

[2] A.Favre, Physics of Fluids, **26**, 2851, (1983)

[3] M. Germano, "Turbulence: the filtering approach," J. Fluid Mech. **238**, 325 (1992).

[4] B. Vreman, B. Geurts and H. Kuerten, "Realizability conditions for the turbulent stress tensor in large-eddy simulation," J. Fluid Mech. **278**, 351 (1994).

[5] A.W. Vreman, B.J. Geurts and J.G.M. Kuerten, "Subgrid-modelling in LES of compressible flows," Direct and Large-Eddy Simulation I, P.R. Voke, L. Kleiser and J.P. Chollet (editors), Kluwer, 133 (1994).

[6] U. Piomelli, T.A. Zang, C.G. Speziale and M.Y. Hussaini, "On the large-eddy simulation of transitional wall-bounded flows," Phys. Fluids A **2**, 257 (1990).

[7] J. Bardina, J.H. Ferziger and W.C. Reynolds, "Improved turbulence models based on LES of homogeneous incompressible turbulent flows," Department of Mechanical Engineering, Report No. TF-19, Stanford (1984).

[8] R.A. Clark, J.H. Ferziger and W.C. Reynolds, "Evaluation of subgrid-scale models using an accurately simulated turbulent flow," J. Fluid Mech. **91**, 1 (1979).

[9] S. Liu, C. Meneveau and J. Katz, "On the properties of similarity subgrid-scale models as deduced from measurements in a turbulent jet," J. Fluid Mech. **275**, 83 (1994).

[10] B. Vreman, B. Geurts and H. Kuerten, "On the formulation of the dynamic mixed subgrid-scale model," Phys. Fluids **6**, 4057 (1994).

[11] B. Vreman, B. Geurts and H. Kuerten, "Large Eddy Simulation of the temporal mixing layer using the Clark model," Memorandum No. 1213, University of Twente (1994).

[12] M.M. Rai and P. Moin, "Direct numerical simulation of transition and turbulence in a spatially evolving boundary layer," J. Comp. Phys. **109**, 169 (1993).

[13] B. Vreman, H. Kuerten and B. Geurts, "Shocks in direct numerical simulation of the confined three-dimensional mixing layer", Physics of Fluids, to appear (1995).

[14] J. Broeze, B. Geurts, H. Kuerten and M. Streng, "Multigrid acceleration of time-accurate DNS of compressible turbulent flow," Copper Mountain (1995).

[15] E.F. van de Velde, "Concurrent Scientific Computing," Springer Verlag, New York (1994).

[16] M. Streng and R. van der Pas, "Implementation of a compressible flow solver on the Power Challenge Array", in preparation.

[17] M. Streng and R. van der Pas, "Some performance considerations for the R8000 Microprocessor", in preparation.

A straightforward 3D multi-block unsteady Navier-Stokes solver for direct and large-eddy simulations of transitional and turbulent compressible flows

P. Comte, J.H. Silvestrini & E. Lamballais
Turbulence Simulation & Modelling Team
L.E.G.I./Institut de Mécanique de Grenoble *
B.P. 53 X, F38041 Grenoble Cedex, France
Tel: (33) 76 82 51 21, Fax: (33) 76 82 52 71, E-Mail: comte@img.fr

1. ABSTRACT

A versatile and effective numerical code for direct and large-eddy simulations of compressible flows is described. It is based on robust explicit finite-difference methods which are second-order accurate in time and fourth-order accurate in space (Gottlieb & Turkel, 1976). An industrial application is presented, with comparison to a more fundamental case, tackled with spectral and compact schemes.

2. INTRODUCTION

Traditionally, the algorithmic concern in CFD has been the (fast) convergence of steady calculations of flows over complex objects. Unsteadiness in the CFD context is generally associated to changes of angle of attack or geometry (in the case of store separation, for example), but scarcely to the Tollmien-Schlichting waves or the hairpin vortices which develop within the boundary layers around these objects. Such events are generally considered as "turbulent fluctuations" and are either ignored or expected to be accounted for through one-point-closure turbulence models. This might yield acceptable prediction of the overall drag over an aircraft, but fails at predicting, for example, the length of the transitional region in a boundary layer subjected to a given level of perturbations. The reason for this is that the physical mechanisms (of transition, turbulence or separation) are not understood at a fundamental level. There is therefore a great need for numerical simulations of transitional, turbulent or separated flows. Considering the computational resources currently available, two strategies are possible:

- Direct Numerical Simulations, in which all turbulent scales are simulated explicitely, in three dimensions of space, down to the Kolmogorov scale η (or nearly so). This implies high-order unsteady schemes, small time-steps, very fine 3D grids and, in practice, low Reynolds numbers. We would like to stress that it is not because a given scheme solves the complete Navier-Stokes equations in three dimensions that its solutions automatically deserve the DNS label. If the mesh size in a turbulent region is larger than, say, 10η, we cannot speak of DNS. Some use the expression *"pseudo-DNS"*. The problem in this case is that the amount of dissipation brought about by the grid being too coarse is not controlled.

- Large-Eddy Simulations, a *half-way house* (Leschziner, 1995) between DNS and one-point closures. It consists of simulating explicitely and in three dimensions all motion larger than a certain cut-off scale, accounting for the contribution of the smaller scales through a simple algebraic model. This presupposes that the large scales are more important than the small ones, which is certainly true for turbulence but is more doubtful for combustion, for example. In any case, from the point of view of algorithmics, the numerical methods used for LES are the same as for DNS, except that the subgrid-scale turbulence model induces non-linearities in the dissipative terms (in addition to those which come from the dependence of molecular viscosity with respect to local temperature).

One controversial question (within the scope of this panel) is the role that can play numerical dissipation in the turbulence-modelling process, either through the nature of the scheme or the mesh size. In our LES, the solutions to the equations solved do contain a certain level of kinetic energy in the smallest resolved scales. This is sometimes criticized on the ground that all numerical methods behave badly in the small scales (even the spectral methods blur the phase information at the highest wavenumber). Validation then has to be performed on physical grounds, through comparison with experimental data, predictions of stability theories or numerical results obtained with different methods. Note that Leonard (1974), who coined the expression *Large-Eddy Simulation*, proposed a formalism thanks to which no energy would be left in the smallest resolved scales.[1] To the other extreme, some claim that numerical dissipation can play the role of a subgrid-scale turbulence model, and sometimes that of the molecular viscous terms too (approaches refered to as *Monotonically Integrated LES*, *Built-In LES*, and so on)

Before giving our point of view, we will briefly recall the very classical numerical methods we use for the simulation of compressible flows which do not develop strong

*Institut National Polytechnique de Grenoble (INPG), Université Joseph Fourier (UJF) et Centre National de la Recherche Scientifique (CNRS).

[1] The Navier-Stokes equations are first convolved through a continuous low-pass filter which commutes with the time and space derivatives. The resulting equations are then closed thanks to a subgrid-scale turbulence model. This closed system of equations is eventually discretized onto a grid which is finer than the cut-off scale of the filter, so that the result can be checked to be independent of the mesh size (but of course not of the filter's cut-off scale).

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

shocks. The subgrid-scale turbulence models that we currently used are briefly presented in section 4, although they will be presented in more details in Lesieur & Métais (1996). The soundest of these models is then applied to the LES of an incompressible mixing layer performed with spectral-like methods renowned for their low numerical dissipation and dispersion. The same model is finally applied to a more industrial mixing layer simulated with the code described below.

3. NUMERICAL SCHEME

In cartesian co-ordinates, the compressible LES equations can be cast, after several crude simplifications discussed in Comte *et al.* (1994), in the conservation-like form

$$\frac{\partial U}{\partial t} + \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2} + \frac{\partial F_3}{\partial x_3} = 0 \quad , \tag{1}$$

with

$$U = \,^T(\rho, \rho u, \rho v, \rho w, \rho e) \quad , \tag{2}$$

in which ρe stands for the resolved total energy defined, for an ideal gas (air), by

$$\rho e = \rho \; c_v \; T + \tfrac{1}{2}\rho(u_1^2 + u_2^2 + u_3^2) \quad . \tag{3}$$

The fluxes F_i read, $\forall i \in \{1,2,3\}$,

$$F_i = \left[\begin{array}{l} \rho u_i \\ \rho u_i u_1 + \rho RT\delta_{i1} - (\mu + \rho\nu_t)\tau_{i1} \\ \rho u_i u_2 + \rho RT\delta_{i2} - (\mu + \rho\nu_t)\tau_{i2} \\ \rho u_i u_3 + \rho RT\delta_{i3} - (\mu + \rho\nu_t)\tau_{i3} \\ \rho(e + RT)u_i - \mu\tau_{ij}u_j - (k + \rho c_p \kappa_t)\dfrac{\partial T}{\partial x_i} \end{array} \right] \tag{4}$$

with $R = \frac{\mathcal{R}}{\mathcal{M}} = 287.06 \; Jkg^{-1}K^{-1}$ and

$$\tau_{ij} = \left[\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} - \frac{2}{3}(\vec{\nabla}.\vec{u})\delta_{ij} \right] \quad , \tag{5}$$

the deviatoric part of the resolved strain-rate tensor. Molecular viscosity is prescribed through Sutherland's law

$$\mu(T) = \mu(273.15)\sqrt{\frac{T}{273.15}} \cdot \frac{1 + S/273.15}{1 + S/T} \quad \forall T \geq 120 \quad , \tag{6a}$$

with $\mu(273.15) = 1.711 \; 10^{-5} Pl$ and $S = 110.4$, and its extension to temperatures lower than 120 K:

$$\mu(T) = \mu(120) \; T/120 \quad \forall \, T < 120 \quad , \tag{6b}$$

$\mu(120)$ being given by eq. (6a). The molecular conductivity $k(T)$ derives from the constant-Prandtl-number assumption $Pr = c_p\mu(T)/k(T) = 0.7$. To be closed, this set of equations requires the definition of ν_t and κ_t, eddy-viscosity and eddy-diffusivity coefficients provided by the SGS model used. This will be done in the next section.

The adaptation to curvilinear co-ordonates was done by David (1993), following Viviand (1974) (see also the complete development in Fletcher, 1988), keeping the spanwise co-ordinate x_3 cartesian. The chain rule gives

$$\frac{\partial}{\partial x_i} = \frac{\partial}{\partial \xi_1}\frac{\partial \xi_1}{\partial x_i} + \frac{\partial}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_i} \tag{7}$$

for any regular co-ordinate transformation $(x_1, x_2, x_3) \longrightarrow (\xi_1, \xi_2, \xi_3 = x_3)$. Introducing the Jacobian

$$J = \det \begin{pmatrix} \dfrac{\partial \xi_1}{\partial x_1} & \dfrac{\partial \xi_1}{\partial x_2} & 0 \\ \dfrac{\partial \xi_2}{\partial x_1} & \dfrac{\partial \xi_2}{\partial x_2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad , \tag{8}$$

one can then re-write (1) as

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial \hat{F}}{\partial \xi_1} + \frac{\partial \hat{G}}{\partial \xi_2} + \frac{\partial \hat{H}}{\partial x_3} = 0 \quad , \tag{9}$$

with

$$\hat{U} = U/J \tag{10a}$$

$$\hat{F} = \frac{1}{J} \; \left[\left(\frac{\partial \xi_1}{\partial x_1}\right) F_1 + \left(\frac{\partial \xi_1}{\partial x_2}\right) F_2 \right] \tag{10b}$$

$$\hat{G} = \frac{1}{J} \; \left[\left(\frac{\partial \xi_2}{\partial x_1}\right) F_1 + \left(\frac{\partial \xi_2}{\partial x_2}\right) F_2 \right] \tag{10c}$$

$$\hat{H} = \frac{1}{J} \; F_3 \quad , \tag{10d}$$

using the chain rule (7) for the derivatives arising in the fluxes $\hat{F}$, $\hat{G}$ and $\hat{H}$. Vector U is still a function of the cartesian co-ordinates x_i and time t. In the limit of zero viscosity and conductivity (Euler equations without SGS model), the fluxes F_i – still defined by (3) – would be functions of U only.

For a given 2D geometry nearly-orthogonal curvilinear grid $\xi_1(x_1, x_2)$; $\xi_2(x_1, x_2)$ is generated by Ryskin method, in such a way that each boundary of the domain corresponds either to a line at constant ξ_1 or at constant ξ_2. This grid is then made 3D by spanwise translation. The system (9) is solved on this grid by means of a (2,4) extension of the fully-explicit McCormack scheme devised by Gottlieb and Turkel (1976), in the form

$$U^{(1)}_{i,j,k} = U^n_{i,j,k} - J^{(p)}_{i,j,k} \left\{ \begin{array}{l} \frac{\Delta t}{\Delta \xi_1}\Big[\;\; \frac{7}{6}\left(\hat{F}^n_{i+1,j,k} - \hat{F}^n_{i,j,k}\right) \\ \qquad\quad -\frac{1}{6}\left(\hat{F}^n_{i+2,j,k} - \hat{F}^n_{i+1,j,k}\right)\Big] \\ +\frac{\Delta t}{\Delta \xi_2}\Big[\;\; \frac{7}{6}\left(\hat{G}^n_{i,j+1,k} - \hat{G}^n_{i,j,k}\right) \\ \qquad\quad -\frac{1}{6}\left(\hat{G}^n_{i,j+2,k} - \hat{G}^n_{i,j+1,k}\right)\Big] \\ +\frac{\Delta t}{\Delta x_3}\Big[\;\; \frac{7}{6}\left(\hat{H}^n_{i,j,k+1} - \hat{H}^n_{i,j,k}\right) \\ \qquad\quad -\frac{1}{6}\left(\hat{H}^n_{i,j,k+2} - \hat{H}^n_{i,j,k+1}\right)\Big] \end{array} \right\} \tag{11a}$$

$$\begin{array}{rl} U^{n+1}_{i,j,k} = & \frac{1}{2}\left[U^{(1)}_{i,j,k} + U^n_{i,j,k}\right] \\ & -\frac{1}{2}J^{(c)}_{i,j,k} \left[\begin{array}{l} \frac{\Delta t}{\Delta \xi_1}\Big[\;\; \frac{7}{6}\left(\hat{F}^{(1)}_{i,j,k} - \hat{F}^{(1)}_{i-1,j,k}\right) \\ \qquad\quad -\frac{1}{6}\left(\hat{F}^{(1)}_{i-1,j,k} - \hat{F}^{(1)}_{i-2,j,k}\right)\Big] \\ +\frac{\Delta t}{\Delta \xi_2}\Big[\;\; \frac{7}{6}\left(\hat{G}^{(1)}_{i,j,k} - \hat{G}^{(1)}_{i,j-1,k}\right) \\ \qquad\quad -\frac{1}{6}\left(\hat{G}^{(1)}_{i,j-1,k} - \hat{G}^{(1)}_{i,j-2,k}\right)\Big] \\ +\frac{\Delta t}{\Delta x_3}\Big[\;\; \frac{7}{6}\left(\hat{H}^{(1)}_{i,j,k} - \hat{H}^{(1)}_{i,j,k-1}\right) \\ \qquad\quad -\frac{1}{6}\left(\hat{H}^{(1)}_{i,j,k-1} - \hat{H}^{(1)}_{i,j,k-2}\right)\Big] \end{array} \right] \end{array} \tag{11b}$$

As mentioned in Thompson *et al.* (1985) and recalled in Fletcher (1988), the metrics $\partial\xi_i/\partial x_j$ arising in the fluxes and Jacobians above have to be discretized in such a way that unwanted cross-terms cancel out, otherwise the scheme is not consistent. First of all, they have to be expressed as analytic functions of the metrics $\partial x_\ell/\partial\xi_m$ of the inverse transform, in order to eliminate all derivatives with respect to x_1 and x_2 in (9) and (10). These inverse metrics are discretized in the following manner:

$$\left(\frac{\partial x_\ell}{\partial \xi_1}\right) = \begin{cases} \dfrac{-1/6\ x_{\ell_{i+2,j,k}} + 8/6\ x_{\ell_{i+1,j,k}} - 7/6\ x_{\ell_{i,j,k}}}{\Delta \xi_1} \\ \text{in the predictor step (11a), and} \\ \\ \dfrac{7/6\ x_{\ell_{i,j,k}} - 8/6\ x_{\ell_{i-1,j,k}} + 1/6\ x_{\ell_{i-2,j,k}}}{\Delta \xi_1} \\ \text{in the corrector step (11b)} \end{cases} \tag{12a}$$

and

$$\left(\frac{\partial x_\ell}{\partial \xi_2}\right) = \begin{cases} \dfrac{-1/6\ x_{\ell_{i,j+2,k}} + 8/6\ x_{\ell_{i,j+1,k}} - 7/6\ x_{\ell_{i,j,k}}}{\Delta \xi_2} \\ \text{in the predictor step (11a), and} \\ \\ \dfrac{7/6\ x_{\ell_{i,j,k}} - 8/6\ x_{\ell_{i,j-1,k}} + 1/6\ x_{\ell_{i,j-2,k}}}{\Delta \xi_2} \\ \text{in the corrector step (11b)} \end{cases} \tag{12b}$$

This is only first-order accurate, which is justified by the fact that the grids we use are not very distorted, except very locally. Therefore $\partial x_\ell / \partial \xi_m$ remains almost everywhere close to $\delta_{\ell m}$.

In the same way, the chain rule (7) has to be applied to eliminate all derivatives with respect to x_1 and x_2 from the fluxes F_i. This introduces metrics to be evaluated as said above, together with derivatives of velocity and temperature with respect to ξ_1 and ξ_2. Consistency then determines the way these derivatives, and also $\partial/\partial \xi_3 \equiv \partial/\partial x_3$, should be discretized.

The boundary conditions are based on a decomposition into characteristics, in the spirit of Thompson (1987, 1990) and Poinsot and Lele (1992). The Riemann invariants of outgoing characteritics are extrapolated, whereas the incoming ones are either prescribed (*e.g.* at the inflow boundary) or set to zero (*non-reflective* or *open* boundary condition). For example, going back to cartesian co-ordinates for the sake of simplicity, in the case of a boundary perpendicular to the direction x_1, the Euler equations are recast in their quasi-linear form

$$\frac{\partial V}{\partial t} + A\ \frac{\partial V}{\partial x_1}\ = 0, \quad \text{with} \quad V = {}^T(\rho, \rho u_1, \rho u_2, \rho u_3, p) \quad . \tag{13}$$

The matrix A is, as per usual, diagonalized in the form $\Lambda = L^{-1}AL$. Assuming L to be locally constant and introducing the vector $W = LV$, system (13) decouples into 5 equations of the form

$$\frac{\partial w}{\partial t} + \lambda\ \frac{\partial w}{\partial x_1}\ = 0 \quad , \tag{14}$$

to be solved at the boundary point N through the semi-implicit scheme

$$\frac{w_N^{n+1} - w_N^n}{\Delta t} + \frac{\lambda_N^n + |\lambda_N^n|}{2} \left[\frac{w_N^{n+1} - w_{N-1}^{n+1}}{\Delta x_1}\right] + \frac{\lambda_N^n - |\lambda_N^n|}{2} \left[\frac{w_{N+1}^{n+1} - w_N^{n+1}}{\Delta x_1}\right] = 0 \quad . \tag{15}$$

For the outgoing characteristics ($\lambda_N^n > 0$), the values of w_N^{n+1} are obtained from that of λ_N^n, w_N^n and w_{N-1}^{n+1}, which are supposed to be known. For the incoming characteristics ($\lambda_N^n < 0$), it is necessary to prescribe w_{N+1}^{n+1} in order to pull out w_N^{n+1}. This is done by considering the nature of the boundary condition (adherence, free slip, periodicity, prescribed flow rate, non-reflectivity, inter-block matching...). V_N^{n+1} is finally deduced from W_N^{n+1} assuming simply $L_N^{n+1} = L_N^n$.

3. RAPID OVERVIEW OF OUR SGS MODELS AND THEIR RECENT EVOLUTIONS

Assuming spectra $E(k) \propto k^{-m}$ for all k, Métais & Lesieur (1992) proposed models defined in the spectral space, reading in a simplified form

$$\nu_t(k,t) = \ 0.31\ \frac{5-m}{m+1}\ \sqrt{3-m}\ \ C_K^{-3/2} \qquad \nu_t^*(k/k_c)\ \sqrt{\frac{E(k_c,t)}{k_c}}\ \text{for}\ m < 3, \tag{16a}$$

and

$$\nu_t(k,t) = 0 \qquad\qquad \text{for}\ m \geq 3, \tag{16b}$$

with

$$\kappa_t(k,t) = \nu_t(k,t)/Pr_t \qquad \text{with} \quad Pr_t = 0.6 \quad . \tag{16c}$$

C_K denotes Kolmogorov's constant, and $\nu_t^* = 1$ for $k/k_c \leq\simeq 0.3$. It rises for higher k/k_c, a good fit of it is (in the case $m = 5/3$ at least),

$$\nu_t^*(k/k_c) = 1 + 34.5 \exp[-3.03\ k_c/k] \quad . \tag{17}$$

Until now, this model has been used with a fixed value $m = 5/3$, giving satisfactory results, not only in the case of isotropic turbulence but also stratified and/or rotating homogeneous turbulence and temporally-growing free shear flows (mixing layers, wakes). For streamwise-and-spanwise-periodic wall-bounded flows, the easiest way of accomodating grid refinement at the wall is to work on xz planes, normal to the wall, over which 2D spectra $E_{2D}(k_{2D}, y, t)$ can be computed. Assuming again isotropy with $E(k) \propto k^{-m}$, one can relate E_{2D} to E and express eddy viscosity and conductivity $\nu_t(k_{2D}, y, t)$ and $\kappa_t(k_{2D}, y, t)$ from (16a). One of us (E.L.) did it in the case of a plane turbulent channel flow. With $m = 5/3$, results are qualitatively correct, but the wall shear stress τ_w are underestimated by about 20% (Fig. 1, top). This is because the model is too dissipative near the wall, where experimental measurements show spectra steeper than $k^{-5/3}$. Much better statistics are obtained with a variable $m(y,t)$ estimated at each timestep from $E_{2D}(k_{2D}, y, t)$ through a least-square fit between $k_{2D_c}/2$ and k_{2D_c}, the cut-off wavenumber (Fig. 1, bottom).

These results correspond to simulations at $R = U_{c_{lam}} h/\nu = 5000$, in which h denotes the channel's half height and $U_{c_{lam}}$ the centerline velocity of a laminar Poiseuille flow of same flow rate (usual convention). This should yield $R_\tau = u_\tau h/\nu \approx 200$, which is the case for the top plot of Fig. 1 (instead of ≈ 180 for the bottom one). Both calculations are performed by means of de-aliased pseudo-spectral methods on xz planes and 6th order compact schemes in the y direction (details will be provided in Lamballais et al., in preparation). The resolution is $64 \times 65 \times 32$, for a domain of size $2\pi h \times 2h \times \pi h$, so that the cut-off wavenumbers along x and z are the same. Extension to non-square

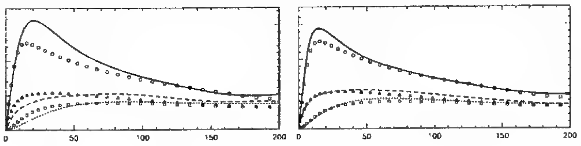


FIG. 1 - u_{rms} (solid), v_{rms} (dotted) and w_{rms} (dashed) obtained from 2 LES differing only in the determination of m: set to 5/3 in the top plot and evaluated from $E_{2D}(k_{2D}, y, t)$, which turned the model off in the viscous sublayer (bottom plot). In both plots, the symbols correspond to LES by Piomelli (1993) with Germano-Lilly's dynamic model, which are very close to experimental measurements.

meshes is in progress; in particular, the procedure proposed in Scotti et al. (1993) is being tested.

When spectral methods cannot be used, we strive to determine eddy-viscosities out of a measure of the kinetic energy at the smallest resolved scale $\Delta = \pi/k_c$. One of these local spectra is $F_{2_\Delta}(\vec{x}, t)$, the second-order structure function of the resolved velocity field, evaluated by averaging over the closest neighbours of point $\vec{x}$, either in all 3 directions of space (6-neighbour formulation) or on planes normal to the wall or mean shear (4-neighbour formulation). In the case of infinite Kolmogorov spectra, energy-conservation arguments (Leslie & Quarini, 1979) yield the *structure-function model* (Métais & Lesieur, 1992), defined by

$$\nu_t^{SF}(\vec{x}, t) = 0.105\ C_K^{-3/2}\ \Delta\ \sqrt{F_{2_\Delta}(\vec{x}, t)} \quad , \qquad (18)$$

consistent with the spectral model (16a).

This SF-model appears to be slightly less dissipative than the Smagorinsky model with the constant 0.18 given by the same assumptions (infinite Kolmogorov cascade, see e.g. Comte et al., 1994). As it involves velocity increments instead of derivatives, it also has the advantage of being defined independently of the numerical scheme used. It is nevertheless not much better for transition than the Smagorinsky model: low-wavenumber velocity fluctuations corresponding to unstable modes yield ν_t's large enough to affect the growth rate of weak unstabilities like Tollmien-Schlichting waves. So far, we have found two ways of remedying this:

- apply a high-pass filter onto the resolved velocity field before computing its structure function. With a triply-iterated second-order finite-difference Laplacian filter denoted $\tilde{\ }$, one finds $\tilde{E}(k)/E(k) \approx 40^3\ (k/k_c)^9$ for all k, almost independently of the velocity field and resolution. With the same arguments as for the structure-function model, this yields the *filtered structure-function model*, defined by (Ducros et al, 1995)

$$\nu_t^{FSF}(\vec{x}, t) = 0.0014\ C_K^{-3/2}\ \Delta\ \sqrt{\tilde{F}_{2_\Delta}(\vec{x}, t)} \quad . \qquad (19)$$

This model enabled Ducros to perform the LES of a spatially-growing boundary layer (at Mach 0.5) between $Re_x = 3.3\ 10^5$ and $1.14\ 10^6$, which widely encompasses the transition region, for a cost of about 80 hours of Cray 2. With the first mesh line at $y^+ \approx 3$ (i.e. with just one point in the viscous sublayer) and only 32 points along y, statistics were found to be within 20% agreement with experimental data, as in Fig. 1 top.

- switch the original structure-function model off when the flow is not three-dimensional enough in the small scales (David, 1993). In practice, an average vorticity vector $\tilde{\vec{\omega}}(\vec{x}, t)$ is computed over $\vec{x}$ and its (4 or 6) closest neighbours. The structure-function model is applied only if the magnitude of the angle $\alpha = (\vec{\omega}(\vec{x}, t), \tilde{\vec{\omega}}(\vec{x}, t))$ exceeds a certain threshold α_0. Simulations[2] of incompressible isotropic turbulence at resolutions ranging between 32^3 and 64^3 gave pdf's of $|\alpha|$ peaking around 20°. Having found the choice of α_0 not critical between 10 and 45°, we finally retained $\alpha_0 = 20°$. The model's constant was finally set to 1.56 times that of the SF model, a least-square fit between our test simulations yielding the same average dissipation as the SF-model. Dispersion was found small enough to justify this in first approximation, but a lot of work has yet to be done to reduce the arbitrariness in this model. In any case, the most surprising conclusion about the filtered and selective structure-function models (hereafter FSF and SSF, respectively) is that they can be interchanged without much difference in the results (Comte et al., 1994). This comes from the fact that they both considerably shrink the support of ν_t (with respect to that of the original SF model), and that both supports are almost the same (Fig. 2, middle and bottom plots). In any case, they do not react to Λ-vortices, whereas the SF model does (Fig. 2 top).

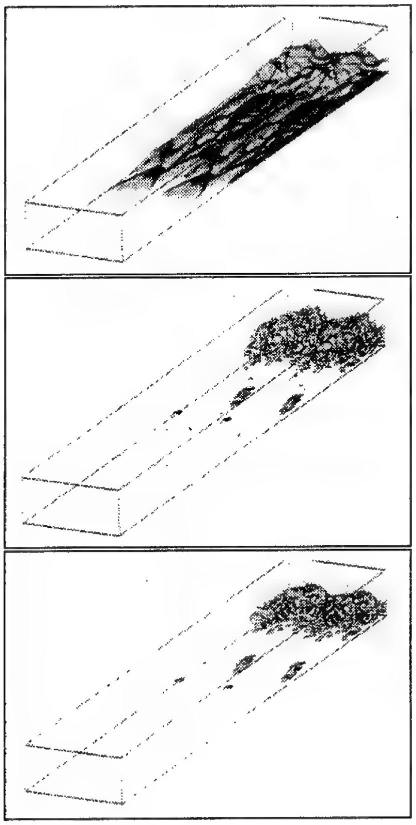


FIG. 2 - From top to bottom: isosurfaces $\nu_t = 2/3\ \nu$ given by the SF, FSF and SFS models, respectively, in the transitional portion of a spatially-growing boundary layer at Mach 0.5 simulated with the FSF model (Ducros and Ducros et al., 1995, or Comte et al., 1994). The same velocity field was used for the three plots (a priori test).

[2] LES with the original structure-function model

4. A REFERENCE CASE: THE INCOMPRESSIBLE MIXING LAYER SIMULATED WITH SPECTRAL-LIKE METHODS

The scheme described above is deliberately dissipative, in order to make it robust: as an example, we will show in the next section an application of it in the case of a solid-propellant booster. Let us also mention the flow over a compression ramp at Mach 2.5 or the boundary layer at Mach 4.5 that we briefly presented at the 74th AGARD FDP (see Comte & David, 1995 and Ducros *et al.* 1993 for more details). The price to pay for this robustness is a certain numerical dissipation (let alone the numerical dispersion), which is difficult to measure. In the absence of really-conclusive analytical arguments, one way around is perform comparisons with results obtained from numerical methods famous for their precision, such as the spectral or collocation methods.

In Comte et al (1992), we presented a comparison between two pseudo-spectral DNS of incompressible mixing layers at Reynolds number[3] 100 differing only by the nature of the initial perturbations. In one case, these were made of a mixture between 2D fluctuations (energy $10^{-4}U^2$) and 3D fluctuations of energy $10^{-5}U^2$. The result was the formation of quasi-2D Kelvin-Helmholtz vortices undergoing pairings and stretching weak hairpin vortices between one another. The spectra measured were in $\approx k^{-3}$ or k^{-4}, even after the second pairing, and vorticity remained bounded by its maximal initial value ω_i. In the other case (same 3D fluctuations as before, but of energy $10^{-4}U^2$ and without 2D perturbulations) *helical pairing* were observed, with more energy in the small scales (spectra in $k^{-5/3}$), and all components of vorticity reaching about 3 ω_i.

These simulations were repeated in LES without molecular viscosity (Silvestrini, 1993). In both case, we observed the same large-scale vortex pattern as in DNS, but with more numerous and intense small-scale vortices ($\max \omega_x \approx 6\omega_i$). The difference in the statistics between the two cases was smaller than in DNS, although the case with 3D perturbations only remained more turbulent.

We now present the same kind of comparison in a spatially-growing configuration, for a velocity ratio $\lambda = (U_1 - U_2)/(U_1+U_2) = 1/2$. Sixth-order accurate compact schemes are used along x with radiative outflow boundary conditions, and pseudo-spectral methods on yz planes assuming periodicity along z (spanwise) and free-slip along y (code written by Gonze, 1993, and recently parallelized on Cray T3D by means of slab ↔ pencil transpositions under PVM). In all cases the computational grid is uniform with cubic meshes. Fig. 3 corresponds to a DNS in a domain of size $L_x = 140\delta_i$, $L_y = 28\delta_i$, $L_z = 14\delta_i$ for a resolution 480 × 96 × 48. The upstream Reynolds number is 100, as in Comte et al (1992), which corresponds approximately to the maximal value permitted at this resolution. The upstream forcing is a mixture between 2D noise on the plane $x = 0$ and noise in the transverse direction y only, of respective energies $\varepsilon_{2D}U^2$ and $\varepsilon_{1D}U^2$ with $\varepsilon_{2D} = 10^{-4}$ and $\varepsilon_{1D} = 10^{-3}$, that is, 3% in turbulent intensity.

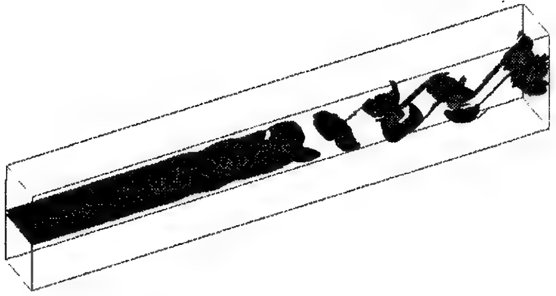


FIG. 3 - Surface $||\vec{\omega}|| = 1/3\ \omega_i$, in DNS à $Re = 100$. Peak vorticity recorded here is 2 ω_i.

Repeated at zero molecular viscosity with the FSF model in its 6-neighbour formulation, transition is obtained farther upstream than before, even with weaker forcing: Fig. 4 is obtained with $\varepsilon_{2D} = 10^{-5}$ and $\varepsilon_{1D} = 10^{-4}$, in a shorter domain than before ($L_x = 112\ \delta_i$, with only 384 points to keep the meshes cubic), other dimensions and number of collocation points being unchanged. The threshold is twice as large as before. Vorticity magnitude now peaks at 4 ω_i, which is compatible with the high-Reynolds number experiments of Huang & Ho (1990). This maximum is reached where streamwise vortices wrap around the primary billows.

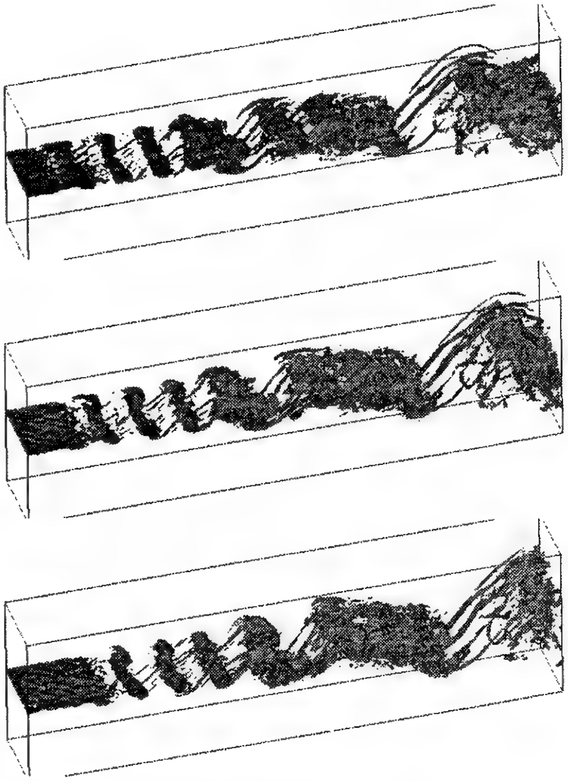


FIG. 4 - Time evolution of the surface $||\vec{\omega}|| = 2/3\ \omega_i$, in a LES at $\nu = 0$, with $\varepsilon_{2D} = 10^{-5}$ and $\varepsilon_{1D} = 10^{-4}$.

[3] based upon $U = (U_1 - U_2)/2$ and δ_i, half the velocity difference and the initial vorticity thickness, respectively

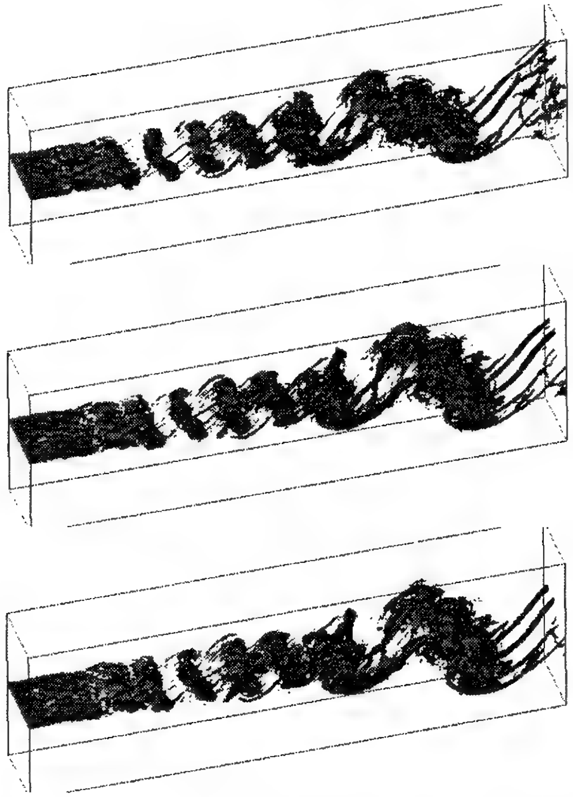


FIG. 4 - (cont'd) - Note the good behaviour of the outflow boundary conditions.

Investigating sensitivity to the nature of the upstream perturbations would not be pertinent in such a narrow domain[4]. We thus doubled L_z and its corresponding number of collocation points. This should not change things much in the quasi-2D case $(\varepsilon_{2D}, \varepsilon_{1D}) = (10^{-5}, 10^{-4})$. However, with $(\varepsilon_{2D}, \varepsilon_{1D}) = (10^{-4}, 0)$, helical pairings are observed in the wider domain (Fig. 5). The interested readers are refered to Silvestrini *et al.* (1995) for more details

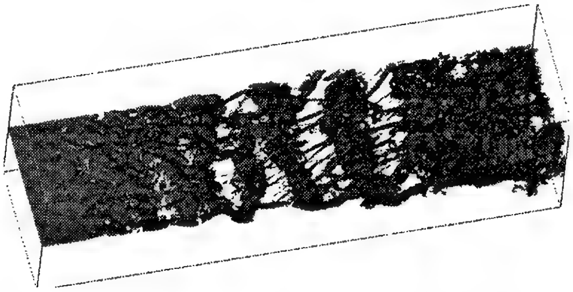


FIG. 5 - Surface $||\vec{\omega}|| = 2/3\ \omega_i$, in LES at $\nu = 0$, with $\varepsilon_{2D} = 10^{-4}$ and $\varepsilon_{1D} = 0$.

5. AN INDUSTRIAL APPLICATION : THE VORTEX SHEDDING INSIDE A SOLID ROCKET ENGINE

We are participating in an operation set up by CNES and ONERA concerning the control of the vibrations induced by vortex shedding within the solid-propellant boosters of the future launcher ARIANE V. We show below preliminary simulations performed with the code described above, in a simplified planar test case, with the grid shown below (Fig. 6).

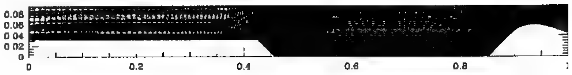


FIG. 6 - Grid of the C1 test case (length $L = 0.47m$, radius $H = 0.045m$, resolution 318 × 31 points

The step is made of burning propellant, at a flame temperature of 3387 K and a mass flow rate, normal to the walls, of 21.2 $kg/m^2/s$. Pressure $p = 4.66\ bar$ is prescribed at the upstream end. The outlet is a nozzle and the outflow boundary conditions are supersonic. The burnt gases are characterized by the following parameters: $\gamma = 1.14$, $R = 299.53\ J/kg/K$, $\mu_{mol} = 9.\ 10^{-5}\ Pl$ et $Pr = 1$.

With such values, 2D simulations are not possible without flux limiters or artificial viscosities. With a viscosity 8 times as large, they become possible without such limiters, and Figure 7 shows the resulting vortices, in time evolution. In such a case, the code gives approximately the same results as the second-order Mc Cormack code SIERRA of ONERA (Lupoglazoff & Vuillot, 1992).

In 3D at the true viscosity and with the filtered structure function model described above, the advantages of the (2,4) scheme become evident. The following figures correspond to a LES at a spanwise resolution of 90 points equally spaced over the span $L_z = \pi\ H \approx 0.141\ m$, with periodic boundary conditions. The initial condition consists of the 2D flow shown above, taken at a given instant of the steady regime, with low-amplitude white noise (of amplitude 10^{-4} the speed of sound at the surface of the propellant) on all the components of $\hat{U}$. Without this perturbation, the flow would have remained 2D, which proves that the code is not "noisy". After having reached the steady regime, which took 50 hours of Cray 90 at 450 Mflops (corresponding to $8ms$ of real time), time se-

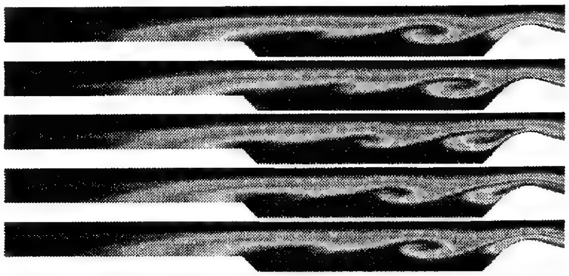


FIG. 7 - Contour maps of entropy at 5 equally spaced instants.

[4] "Spanwise correlation lengths are of the order of $3 - 5\ \delta_\omega$ (δ_ω is the local vorticity thickness). However, the large vortices typically have lengths of order 20 δ_ω when the irregularities along the span are ignored" (quoted from Browand & Troutt, 1985).

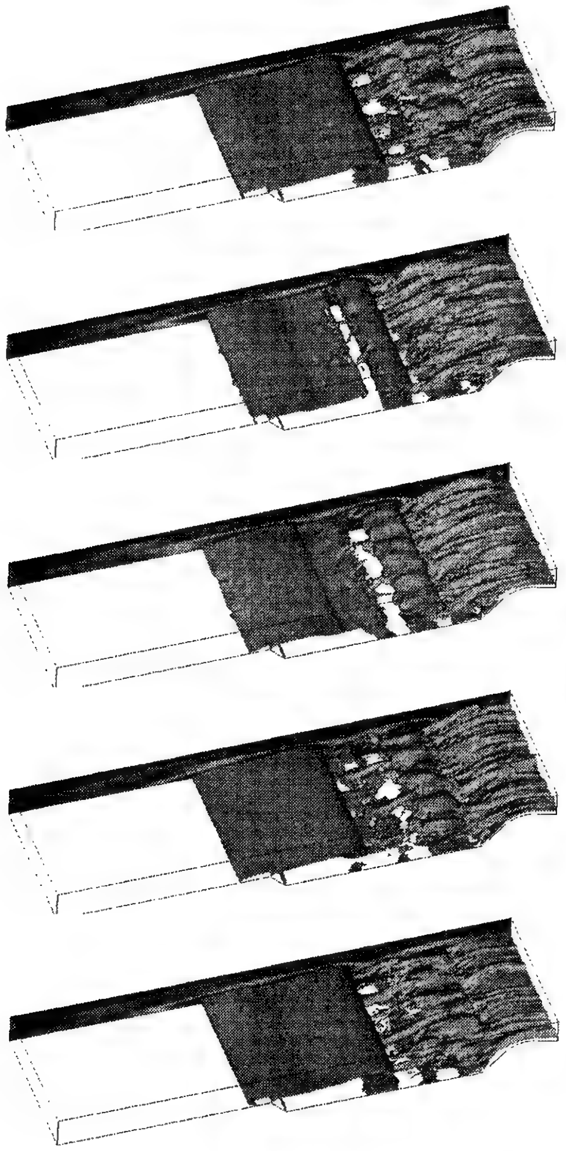


FIG. 8 - Streamwise vortices in a quasi-industrial configuration

ries are recorded for $5ms$. Figure 8 shows an animation of an isosurface of the magnitude of the vorticity vector. Streamwise vortices are not only visible inbetween the large Kelvin-Helmholz billows, as in the previous section, but also at the wall of the nozzle. These are likely to result from a Dean-Görtler instability of the detached boundary layer, which re-attaches in the convergent part of the nozzle (Fig. 9).

The statistics are in global agreement with the experimental data. In particular, we found kinetic energy and pressure spectra which exhibit a fundamental peak around 2500Hz, and its successive harmonics. More precisely, Figure 10 shows a comparison between the present LES and the 2D calculation just above. In the 3D case, the fundamental frequency is lower (2300Hz versus 2670) and the spectra are more developed, in particular in the low frequency. This is of crucial importance for the design of the anti-vibration protections of the rocket's control systems, and illustrates the importance of taking three-dimensionality into account, even when the largest vortices are expected to be two-dimensional.

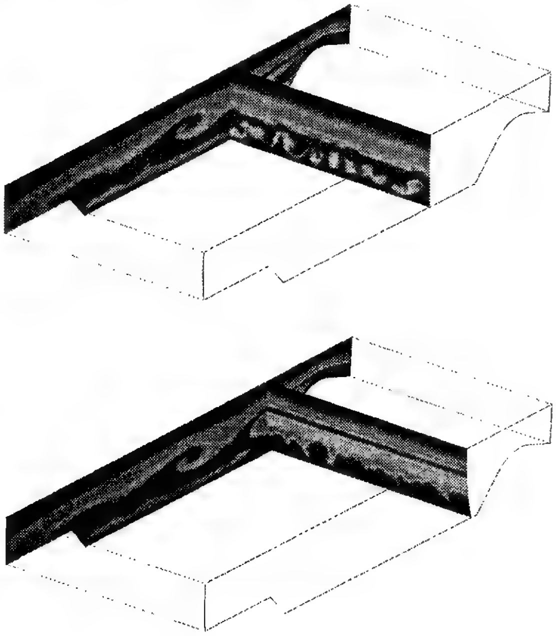


FIG. 9 - Maps of the entropy field. The top view shows a cross section of the Görtler vortices, the bottom one the streamwise vortices which connect the KH billows.

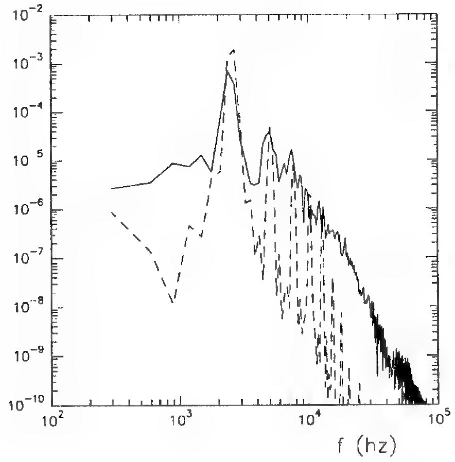


FIG. 10 - Temporal kinetic energy spectra recorded in the middle of the booster. The solid line corresponds to the LES and the dashed line to the 2D DNS.

6. CONCLUSION

A progress report of our efforts towards the industrialization of Large-Eddy Simulations has been presented. In particular, it is shown that such simple algorithms as 5-point extensions of fully-explicit McCormack schemes can be very effective, and compete with spectral methods as far as the description of fine vortical structures is concerned. The importance of longitudinal Görtler-type vortices has been shown in the case of the flow within a simplified booster of Ariane V, in addition to the more dramatic case of HERMES' body flap which was presented orally. The academic simulation of incompressible mixing layers has proved the sensitivity of LES to the nature of the disturbances superimposed onto the basic flow, showing that LES could be a good tool for receptivity studies in aircraft and aerospace research. The next step of our developments in this direction will deal with the adaptation of our subgrid-scale turbulence models to complex geometries, following the footsteps of the Center for Turbulence Research in Stanford (see *e.g.* Ghosal & Moin, 1995). Finally, our opinion about the role that numerical dissipation should play in the turbulence-modelling process is the following: the role of numerical dissipation should be minimized, unless we have a way of controlling it on physical grounds. Algorithms with non-linear dissipation are available: for example, the PPM scheme of Collela & Woodward (1984) is capable of satisfying the second principle of thermodynamics and the positivity of the thermodynamical variables with an amount of numerical dissipation close to the minimum wherever it is not needed. We disagree with the claim that Euler-PPM calculations are LES (either MILES or BILES), because the physics of turbulence has not been incorporated yet. However, we think that this should be possible. Firstly, in subsonic regions at least, its dissipation should be made as little dependent on the grid orientation as possible. Then, we should try to force this dissipation to equal the value prescribed by a given subgrid-scale turbulence model. Thus, explicit eddy-viscosity models might become redundant one day. However, we think that the molecular viscous terms should be kept in all simulations of wall-bounded flows.

7. ACKNOWLEDGEMENTS

Most of the computational time used for the 3D calculations presented here has been freely allocated by IDRIS, the CNRS computing center. The study of the vortex shedding in the boosters of Ariane V is under the CNES/ONERA contract $n°$ 22.492/DA/A1.CC1. The discussion about the role of numerical dissipation has greatly benefitted from the lectures given by Prof. Ferziger, from Stanford University, during his stay in Grenoble last year.

8. REFERENCES

Collela, P. & Woodward, P.R. 1984 The piecewise parabolic method (PPM) for gas-dynamical simulations. *J. Comp. Phys.*, **54**, 174–201.

Comte, P & David, E., 1995, Large-Eddy Simulation of Görtler vortices in a Curved Compression Ramp. *Proc. of the* Second French Russian workshop on experimentation, modelization, computation in flow turbulence and combustion, INRIA/GAMNI-SMAI, eds: B.N. Chetverushkin, A. Desideri, Yur A. Kuznetsov, J. Periaux & B. Stoufflet, John Wiley & sons Publishers (sous presse).

Comte, P. , Ducros, F., Silvestrini, J., David, E., Lamballais, E., Métais, O. & Lesieur, M., 1994, "Simulation des grandes échelles d'écoulements transitionnels", AGARD 74th Fluid Dynamics Panel Meeting and Symposium on "Application of direct and large eddy simulation to transition and turbulence", Chania, Crete, Grèce, 18-21 Avril, 14.1–14.12.

David, E. 1993 *Modélisation des écoulements compressibles et hypersoniques : une approche instationnaire.* Thèse INPG.

Ducros F. 1995 *Simulations numériques directes et des grandes échelles de couches limites compressibles. Thèse* INPG.

Ducros, F., Comte, P. & Lesieur, M., 1995, "Large-eddy simulations of transition to turbulence in a weakly-compressible boundary layer over a flat plate", submitted to *J. Fluid Mech.*.

Fletcher, C.A.J. 1988 Computational techniques for fluid dynamics 2", Springer series in Computational Physics", pp. 484.

Ghosal, S., Lund, T.S. & Moin, P. 1992 A local dynamic model for large eddy simulation. *Annual Research Briefs,* 1992, Center for Turbulence Research, Ames Research Center and Stanford University, pp. 3-25.

Ghosal, S. & Moin, P. 1995 The basic equations for the large-eddy simulation of turbulent flows in complex geometry. *J. Comp. Phys.*, **118**, 24–37.

Gonze, M.A. 1993 *Simulation numérique des sillages en transistion à la turbulence.* Thèse INPG.

Gottlieb, D., & Turkel, E. 1976 Dissipative two-four methods for time-dependent problems. *Math. Comp.*, **30** (136), 703–723.

Huang, L. & Ho, C. 1990 Small-scale transition in a plane mixing layer, *J. Fluid Mech.*, **210**, 475–500.

Leonard, A. 1974 Energy cascade in large-eddy simulations of turbulent flows, *Adv. Geophys.*, **18A**, 237–248.

Lele, S. 1990 Compact finite differenceschemes with spectra-like resolution, *J. Comp. Phys.*, **103**, 16–42.

Leschziner, M. 1995 SMAI-CNRS course on *Numerical Methods for Turbulence*, Orsay, France, June 7–9, 1995.

Lesieur, M., Métais, O. 1996 New trends in large-eddy simulations of turbulence, *Ann. Rev. Fluid Mech.*, **28** (to appear).

Leslie, D.C., Quarini, G.L. 1979 The application of turbulence theory to the formulation of subgrid modelling procedures. *J. Fluid Mech.*, **91**, pp. 65–91.

Lupoglazoff, N. & Vuillot, F. 1992 Numerical simulation of vortex shedding phenomenon in 2D test case solid rocket motors. *AIAA Paper92-0776*, 30th AIAA Aerospace Sciences Meeting Reno, USA

Moin, P. & Kim, J. 1985: The structure of the vorticity field in turbulent channel flow. Part 1. Analysis of instantaneous fields and statistical correlations *J. fluid Mech.*, **155**, pp. 441–464.

Poinsot, T.J. & Lele, S.K. 1992 Boundary conditions for direct simulations of compressible viscous reacting flows, *J. Comp. Phys.* **101**, 104–129.

Scotti A., Meneveau C., Lilly D.K. 1993 Generalized Smagorinsky model for anisotropic grids. *Phys. Fluids A.* **5**, (9), pp. 2306–2308.

Silveira-Neto, A., Grand, D., Métais, O., Lesieur, M. 1993 A numerical investigation of the coherent vortices in turbulence behind a backward-facing step. *J. Fluid Mech.*, **256**, pp. 1–25.

Silvestrini, J. 1993 *Sensibilité aux conditions initiales des structures cohérentes dans une couche de mélange temporelle.* DEA de l'INPG.

Thompson, J.F., Warsi, Z.U.A & Mastin 1985 *Numerical grid generation, foundations and applications*, North-Holland, Amsterdam.

Thompson, K.W. 1987 Time-dependant boundary conditions for hyperbolic systems, *J. Comp. Phys.* **68**, 1–24.

Thompson, K.W. 1990 Time-dependant boundary conditions for hyperbolic systems II, *J. Comp. Phys.* **89**, 439–461.

Silvestrini, J.H., Comte, P. & Lesieur, M., 1995, DNS and LES of incompressible mixing layers developing spatially *Proceedings of Turbulent Shear Flows 10*, Pennsylvania

Viviand, H. 1974 Formes conservatives des équations de la dynamique des gaz, *Rech. Aeros.*, **1**, 65–68.

APPLICATIONS OF LATTICE BOLTZMANN METHODS TO FLUID DYNAMICS

S.A. Orszag
Fluid Dynamics Research Center, Forrestal Campus,
Princeton University, NJ 08544-0710, USA

Y. H. Qian
Department of Applied Physics,
Columbia University, New York, NY 10027, USA

S. Succi
IBM European Center for Scientific and Engineering Computing,
171 P.le Giulio Pastore, Roma, I-00144, Italy

SUMMARY

In this paper, we present recent developments in the theory and application of lattice Boltzmann techniques and related lattice BGK models. Lattice based methods allow the study of complicated systems with simple, efficiently computable physical models. Here we will report some progress with these methods and give an overview of their basic ingredients. Applications to various types of turbulent flows are described.

1 INTRODUCTION

The Lattice Boltzmann Equation (LBE) is a direct method to solve the Navier-Stokes equations on a digital computer. LBE is rooted in boolean lattice gas techniques, a sort of "minimal" molecular dynamics scheme based on the observation that the large-scale dynamics of fluid flow is largely independent of the details of the underlying microdynamics. This suggests that in order to numerically integrate the differential equations describing the motion of a fluid, it may prove convenient to use a population of microvariables ("particles") whose microdynamics can be freely adjusted to match the Navier-Stokes equations on a macroscopic scale [5].

The LBE method takes this approach one step-forward towards the macroscopic world, from the molecular to the kinetic level, by replacing the boolean microdynamical variables with their corresponding floating point expectation values. This move, while preserving the locality in space and time of the evolution rules, which are key to the amenability to parallel computing, offers three main advantages: a better amenability to present-day computing architectures (increasingly faster on the floating point side); a wider degree of latitude in choosing the details of the evolution rule; a reduction of the separation in scale between the micro-world and the macro-world (i.e. the averaging operation on a suitable region of the micro-dynamical lattice needed in boolan simulations to remove statistical noise is no longer necessary).

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

The Lattice Boltzmann equation was introduced in the late 80's to cope with the two major drawbacks of the Lattice Gas Cellular Automata (LGCA) technique: statistical noise and exponential complexity of the evolution rule with the number of degrees of freedom per lattice site.

Ever since, the method has gone from strength to strength up to the point where it can be put on a par with most advanced computational fluid dynamics (CFD) techniques for a large variety of problems, ranging from fully-developed homogeneus incompressible turbulence, to multiphase flows in porous media.

Besides its amenability to parallel computing, the method is appreciated for the ease of implementation of grossly irregular geometries as well as for the flexibility of the evolution rule which allows to model complex physics by minor modifications of the basic collisional scheme.

Despite these brilliant features, LBE has not yet penetrated the CFD engineering community, the primary hurdle being its inability to deal with non-uniform, irregular mesh distributions.

This problem has been partially alleviated in the recent past by importing finite-volume techniques within LBE so as to produce a finite-volume LBE capable of dealing with non-uniform (structured) grids.

This paper is organized as follows: first we present a cursory view of the LGCA nd LBE techniques respectively. Subsequently we describe two applications of LBE to the area of fluid turbulence: three-dimensional Rayleigh-Benard convection and three-dimensional channel flow turbulence.

2 Lattice Gas dynamics

The development of the lattice Boltzmann equation (LBE) is intimately related to lattice gas cellular automata (LGCA). Interest in LGCA originated with the seminal paper of Frisch, Hasslacher & Pomeau (1986) in which it is shown that a simple automaton living on a 2D hexagonal lattice can provide, in the limit of large scale motion, a faithful representation of 2D fluid dynamics [5]. In contrast to the 2D case, no 3D Bravais lattice exists with enough symmetries to lead to 3D fluid dynamics. A clever way out of this problem was found by d'Humières, Lallemand & Frisch (1986) who pointed out that a suitable four dimensional lattice, the face-centered hypercubic (FCHC) lattice, leads to the proper symmetries. To obtain three (two) dimensional hydrodynamics, periodic boundary conditions are imposed along the x_4 direction and the flow is projected into 3D (2D) [3].

The path leading from LGCA to the Navier–Stokes equations is based on a standard procedure of statistical mechanics: (1) to get from the particle level to the Liouville level, an ergodic assumption is used; (2) to get from the Liouville level to the Boltzmann kinetic level, the assumptions that collisions are instantaneous and localized in space are involved; (3) to get from the Boltzmann level to the Navier–Stokes continuum level, the assumption that the particle mean-free-path is much smaller than any macroscopic variation length is made. The formal procedure to achieve the hydrodynamic description of LGCA is based on a multiscale formalism using the Knudsen number as a small parameter.

The main advantages of LGCA are as follows:

- Round-off error-freedom
- Regular data structures, ideal for vector processing
- Local interaction model, ideal for parallel processing
- Ease of implementation of extremely irregular geometries and boundary conditions

The price to be paid for these advantages reflects in the following disadvantages:

- Statistical noise
- Exponential complexity of the collision operator with increasing number of states/site
- Relatively high-viscosity and therefore low effective Reynolds numbers

The issue of statistical noise is a common feature of all particle models; substantial space/time averaging is required to extract reasonably smooth hydrodynamic signals out of the LGCA microdynamics. The issue of exponential complexity is also typical of finite-state algorithms.

3 Lattice Boltzmann dynamics

Lattice Boltzmann techniques provide a way out of both of these problems. With the assumption of molecular chaos, it is possible to write the following kinetic equation:

$$N_i(\vec{x}+\vec{c}_i, t+1) - N_i(\vec{x}, t) = \Delta_i(N) \quad i = 1, b \quad (1)$$

Here $N_i(\vec{x}, t)$ is the ensemble averaged number density of particles of type i lying at the lattice point at $\vec{x}, t$ and propagating along the direction identified by the discrete speed $\vec{c_i}$. Also, $\Delta_i(N)$ is obtained from the boolean collision term by simply replacing the stochastic boolean population n_i with the ensemble averaged population N_i. The problem of noise in equation (1) is absent because N_i is a real variable and no average at all is needed to recover the macroscopic fields. McNamara & Zanetti (1988) proposed to use Eq. (1) directly for hydrodynamic simulations with the Δ_i arising from the corresponding boolean models. In particular, they studied the model defined by the FHP-III rules by simulating the decay of shear and sound waves of finite wavelengthi [10]. The comparison between the numerical values and the Chapman-Enskog multiscale predictions shows that the hydrodynamic value is accurate to better than 5% even for a lattice as small as 4. Also the behavior of sound waves is satisfactory.

The McNamara-Zanetti approach, while fixing the problem of statistical noise, is still left with the intractable complexity of the collision operator because all b-body interactions included in the boolean collision term are still present. This makes their approach unviable in more than two dimensions.

Higuera & Jimenez (1989) [6] noticed that the Lattice Boltzmann equation can be further simplified without losing any generality in terms of hydrodynamic fidelity. The reason is that macrodynamic equations in LGCA formally arise in the double limit of small Knudsen numbers and small

Mach numbers. It is then convenient to consider the expansion of the collision term on the right side of (1) corresponding to these conditions. To do this, let us write N_i as

$$N_i = N_i^{eq}(\rho, \vec{\mathbf{u}}) + N_i^{neq}(\nabla\rho, \nabla\mathbf{u}), \qquad (2)$$

and further decompose N_i^{eq} as

$$N_i^{eq} = N_i^{(0)} + N_i^{(1)} + N_i^{(2)} + O(M^3) \qquad (3)$$

where the upper index refers to the order in the Mach number M. This expansion permits to express the collision operator in terms of a simple 2-body scattering matrix

$$\Delta_i(N) \simeq A_{ij}(N_j - N_j^{eq}). \qquad (4)$$

where $A_{ij} \equiv \frac{\partial\Delta_i}{\partial N_j}$, the derivatives being calculated in the state of zero velocity $N_i = d = \rho/b$. The element A_{ij} controls the scattering rate between directions i and j, and N_i^{eq} is the local maxwellian equilibrium expanded to second order in the local flow field.

Despite its apparent linearity, the expression (4) accounts for second order terms in the expansion of the collision operator.

The Higuera-Jimenez LBE marks an important breakthrough as it opens the way to practical three-dimensional simulations of fluid flows; as a matter of fact it turns a 2^b complex problem (where b is the number of bits at each lattice site) into a b^2 complex one! The quasilinear LBE introduced by Higuera & Jimenez is still in a one–to–one correspondence with its underlying LGCA microdynamics. This sets a relatively strict upper bound to the Reynolds number attainable since the LBE viscosity is exactly the same that results from the corresponding LGCA. Given the fact that one is ultimately interested just in the large–scale, hydrodynamic features of the flow, at this point, this appears as an unnecessary restriction.

One is therefore naturally led to regard the LBE as a *self–standing* model of the Navier–Stokes equations, regardless of any underlying LGCA dynamics (Higuera, Succi, Benzi (1989)) [7].

The starting point in the definition of the 'self-standing' lattice Boltzmann equation is again the linearized kinetic equation (4). The change in perspective is however substantial: the choice of the quantities A_{ij} and N_i^{eq} in (4) is no longer dictated by an underlying boolean microdynamics but is rather adjusted to the macroscopic equations to be reproduced. With this broader view, the attention is shifted on the scattering matrix and notably on its leading non–zero eigenvalue, the one controlling the viscosity of the LBE flow. This eigenvalue can be tuned at the outset so as to achieve the desired flow viscosity in a fairly handy fashion.

4 Lattice BGK models

In a similar vein, Bhatnagar, Gross & Krook (1954) used a relaxation approximation to model the effect of complicated collisions [2]. The basic formulation of lattice BGK models can then be described as a simplified Boltzmann equation starting from time evolution equation as

$$N_i(\vec{x} + \vec{c}_i, t+1) = N_i(\vec{x}, t) + \omega[N_i^e(\vec{x}, t) - N_i(\vec{x}, t)] \qquad (5)$$

where ω is a relaxation parameter (collision frequency in kinetic theory). The key point here is the choice of the equilibrium state N_i^e so that it leads to the exact Navier–Stokes equation at hydrodynamic space and time scales. The right choice is

$$N_i^e = \rho t_p[1 + \frac{c_{i\alpha}u_\alpha}{c_s^2} + \frac{u_\alpha u_\beta}{2c_s^4}(c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta})]$$

where c_s is the speed of sound, and t_p are weights depending on the square amplitude of the velocity p (since particles are either at rest or move one grid site per timestep, p is an index from 0–2 in 2D, 0–3 in 3D, which labels particles at rest, in motion along or in motion diagonal to the grid). Requirements of isotropy and Galilean invariance impose constraints on the weights t_p which are model dependent (Qian and Orszag (1993) [11]).

A two–scale analysis in time leads to the effective hydrodynamic equations at second order of the Knudsen number (the ratio of mean free path to characteristic length):

$$\begin{aligned} \partial_t\rho + \partial_\alpha(\rho u_\alpha) = 0 \\ \partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\alpha u_\beta) = \quad & -\partial_\alpha(c_s^2\rho) \\ +\nu\partial_\beta[\rho(\partial_\beta u_\alpha + \partial_\alpha u_\beta)] \end{aligned} \qquad (6)$$

where c_s is the sound speed and ν the shear viscosity is given by

$$\nu = \frac{c_s^2}{2}(\frac{2}{\omega} - 1)$$

Also, the incorporation of an eddy viscosity model is quite straightforwardly accomplished through the introduction of a space– and time–dependent relaxation parameter ω. The Smagorinsky formula for the eddy viscosity, for example, becomes simply

$$\omega = \frac{2C_1|S| + c_s^2}{2c_s^2} \qquad (8)$$

where $|S|$ is the amplitude of the strain tensor and C_1 is a constant.

A nice property of LBE is that the strain tensor $S_{\alpha\beta}$ is available locally as an appropriate linear combination of the particle populations N_i Other eddy viscosity models may be implemented in a similar way. The inclusion of standard wall conditions for the eddy viscosity is equally straightforward.

From a numerical point of view the LBE is basically an explicit finite–difference scheme working at the edge of the Courant–Friedrichs–Lewy condition $c\Delta t = \Delta x$ and bearing a significant resemblance with the Dufort–Frankel scheme. It is characterized by a favorable computation/calculation ratio which is key to its amenability to parallel implementations across virtually the whole spectrum of present–day parallel computers. This favorable ratio is achieved at the expense of some extra–memory and CPU overhead (the number of discrete populations exceeds the number of significant hydrodynamic fields) as compared to standard explicit CFD schemes.

5 Applications

Many applications of lattice BGK methods to diverse fluid flows have and are being made; for a recent review see (Qian, Succi and Orszag, 1995 [12]).

Here we shall cursorily review two recent applications: three-dimensional Rayleigh-Benard thermal convection and three-dimensional channel flow turbulence.

5.1 Three-dimensional thermal convection

Recently, the LBE formalism has been extended in such a way as to handle thermal convection by including the dynamics of a temperature field within the fluid flow, Massaioli, Succi, Benzi, (1993) [8].

The thermal LBE code has been extensively exploited to gain new insights into a number of issues related to thermal turbulence, such as the shape of the probability distribution function of velocity and temperature fluctuations and the related implications on the scaling properties of thermal turbulence.

Perhaps, the most valuable outcome of these simulations is a clue on the nature of turbulent flows which goes now by the name of "extended Self Similarity" (ESS). ESS represents a kind of generalized scale invariance which apparently holds also in the limit of low Reynolds numbers, i.e when dissipation still plays a non-negligible role on the flow dynamics (Benzi, Ciliberto, Massaioli, Tripiccione and Succi, 1993) [4].

The basic statement of ESS is that scaling properties of a turbulent flow are most conveniently highlighted by inspecting the structure functions one versus another rather than as a function of the space separation r, as suggested by the common practice.

In particular the scaling exponents a_p can be derived by measuring the $p-th$ order distribution function $S_p(r)$ in terms of $S_3(r)$ according to the following relation:

$$S_p(r) = (S_3(r))^{a_p} \tag{7}$$

where $S_p(r)$ is defined as:

$$S_p(r) =< |u(x+r) - u(x)|^p >$$

angular brackets denoting ensemble-averaging.

According to the standard K41 Kolmogorov theory, in the scaling regime (Reynolds number going to infinity) the 3rd order structure function $S_3(r)$ becomes a linear function of the space separation r, which is why scaling is commonly probed by log-plotting the structure functions S_p versus r. The problem with this procedure is that the Reynolds numbers achievable by direct simulation of the Navier-Stokes equations on present-day computers are not high enough to attain a fully-developed scaling regime. The result is that a clearcut measurement of the scaling exponents is hampered by statistical inaccuracies.

The point of ESS is that the eq. (7) holds even for moderately low Reynolds number for which the Kolmogorov relation $S_3(r) \simeq r$ does *not* apply, whence the denomination of "extended" self-similarity.

The practical implication is that scaling exponents can then be reliably measured out of moderate-Reynolds number simulations, well within reach of present-day computational capabilities (128^3 grid points being perfectly adequate).

The validity of the ESS assumption is currently being explored for a variety of different flows, including Rayleigh-Benard turbulence, magnetohydrodynamics and others.

In the specific instance of Rayleigh-Benard convection, ESS has permitted to gather a wide body of numerical evidence in favor of 'buoyancy-driven' Bolgiano scaling , i.e. energy spectrum scaling like $E(k) \simeq k^{-11/5}$, as opposed to 'non-thermal' Kolmogorov decay $E(k) \simeq k^{-5/3}$ [8].

5.2 Three-dimensional turbulent channel flow

As mentioned in the introduction, the LBE has been recently merged with the finite volume method to produce a variant of LBE (FVLBE hereafter) able to deal with non-uniform grids.

The idea is to take the differential form of LB dynamics and apply a finite-volume procedure based upon integration of eq. (1) on each cell of a control grid of (almost) arbitrary shape.

By straigthforward use of Gauss theorem, we obtain

$$\partial_t F_{i,c} + \Phi_{i,c} = \Delta_{i,c} \tag{8}$$

where $F_{i,c}$ is the mean population of the macrocell c, $\Phi_{i,c}$ the corresponding flux across the boundaries of c, and $\Delta_{i,c}$ is the rate of change if $F_{i,c}$ due to collisions occourring within the cell c. Clearly, the actual computaion of surface fluxes involve an interpolation technique. For the case in point, piece-wise linear interpolation is used so that locality is preserved to a good extent in the numerical scheme.

This scheme has been validated for the case of three dimensional turbulent channel flow simulation on a moderate resolution grid ($64 \times 64 \times 128$) spanning a physical channel of heigth $H = 192$, length $L_x = 960$, and width $L_y = 512$, i.e. pretty close to the one examined by Moin and coworkers [9].

The idea is to reproduce the well-known logarithmic law-of-the wall of the mean flow profile:

- $\bar{u}_x(z) = z\frac{v_*^2}{\nu};\ \ 0 \leq z \leq \delta$
- $\bar{u}_x(z) = \frac{v_*}{\chi} log\left(\frac{v_*}{\chi}\right) + v_* d;\ \ z \geq \delta$

where $\chi = 0.4$ is the Von Karman constant, v_* a typical turbulent velocity, d is a calibration constant, and $\delta = \nu / v_*$ is the thickness of the "viscous sublayer".

The average velocity profiles drawn from the numerical simulation are checked against the above expressions to produce best fit values of ν^n, v_*^n, d_n where the superscript n denotes 'numerical simulation' (see Figure 1).

The actual values of ν^n, v_*^n, d_n are derived from the slope of the linear plot $\bar{u}_x\ vs\ z\ (v_*^2/\nu)$, the slope of the plot $\bar{u}_x\ vs\ log(z)$ (v_*/χ) and the value of $log(\bar{u}_x)$ at $z = 1$ $(v_*/\chi \cdot log(v_*/\nu) + dv_*)$.

The main outcome of these simulations is that turbulence is supported during the entire life span of the simulation, that is 2.4×10^5 time steps, corresponding to about 90 longitudinal transit times. This is due to the fact that the channel is long enough to support streamwise rolls feeding cross channel turbulence.

Data samples have been collected every 53 steps in the interval $[100,000, 240,000]$, thus yielding about 2600 profiles for statistical data analysis. The numerical best-fit values deduced from the simulation are as follows: $\nu^n = 0.013 \pm 0.002$, $v_*^n = 0.013 \pm 0.001$, $d^n = 6.5 \pm 0.7$.

First, we remark that the measured viscosity is about twice higher than the theoretical input value $\nu = 0.05$. This is attributed to localized peaks of numerical viscosity occurring there where the lattice pitch is changing due to the mesh non-uniformity (a sharp 1-2-1 mesh distribution along z has been adopted).

Second, we note that d^n is within the error bars provided by the literature, although somewhat on the upper side. Finally, since turbulence is sustained for a significant time-span, wall stress-tensor statistics is also available for the purpose of internal consistency checks This yields:

$$v^* \equiv \sqrt{< u_x u_z >}|_{z=0} \sim 0.012 \qquad (9)$$

in a pretty good match with the values deduced by the velocity profiles (Figure 2).

To sum up, these moderate resolution runs suggest that the FVLBE scheme provides results well within the error bars of current CFD at quite a comparable computational cost (10 μs per grid-point per step on a IBM RS 6000 mod. 580 workstation).

Further work is needed to judge upon its competitiveness on a more quantitative ground.

6 Conclusions

In summary, Lattice Boltzmann methods provide a complementary numerical approach to traditional numerical methods for complex nonlinear systems. Benchmark problems have validated the approach as a flexible and efficient numerical method. Fruitful applications are being made to multiphase flow simulations, subgrid modeling of turbulence and non–uniform lattice applications.

References

[1] G. Amati, S. Succi and R. Benzi, *Fluid Dyn. Res.*, in press

[2] P. Bhatnagar, E.P. Gross, and M.K. Krook, *Phys. Rev.*, 94:511, 1954.

[3] D. d'Humières, P. Lallemand, and U. Frisch, *Europhys. Lett.*, 2:291, 1986.

[4] R. Benzi, S. Ciliberto, F. Baudet, F. Massaioli, S. Succi and R. Tripiccione, *Phys. Rev. E, R29,* , 48, n.1, 1993

[5] U. Frisch, B. Hasslacher, and Y. Pomeau, *Phys. Rev. Lett.*, 56:1505, 1986.

[6] F.J. Higuera and J. Jimenez, *Europhys. Lett.*, 9(7):663–668, 1989.

[7] F. Higuera, S. Succi, R. Benzi *Europhys. Lett.*, 9-4, 1989.

[8] F. Massaioli, R. Benzi, S. Succi and R. Tripiccione, *Eur. J. Mech. B/Fluids*, 14, n.1, 67-74, 1995

[9] P. Moin, J. Jimenez, *J. Fluid Mech.*, 225, 213 1991

[10] G. McNamara and G. Zanetti, *Phys. Rev. Lett.*, 61:2332, 1988.

[11] Y. H. Qian and S. Orszag, *Europhys. Lett.*, 21-3, 1993.

[12] Y. H. Qian , S. Succi and S. Orszag, *Annal Review of Comput. Physics*, vol. 5, 1995, in press

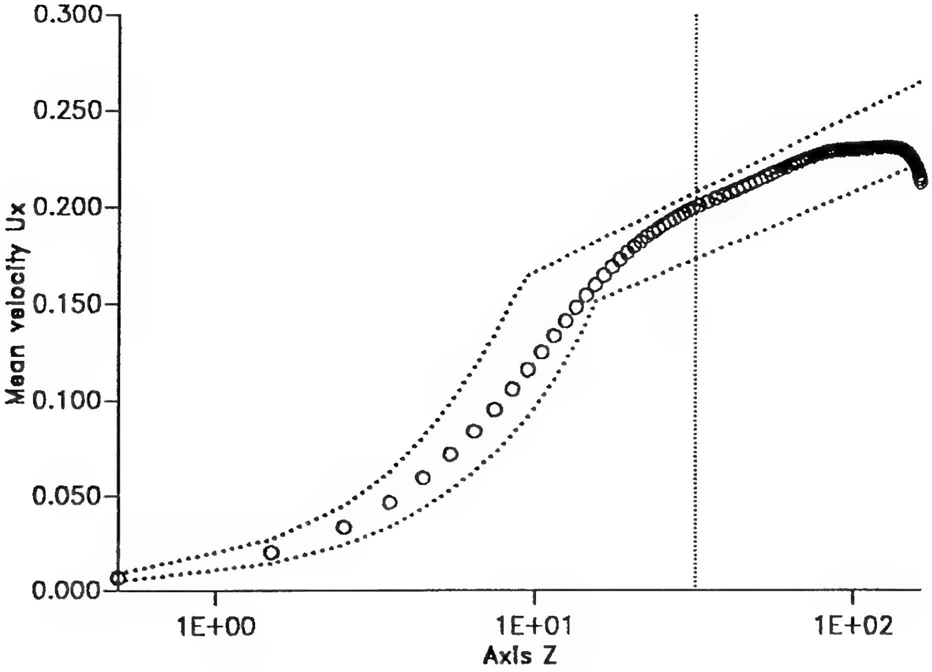


Figure.1 , Mean velocity profile for a turbulent channel flow using moderate resolution, with a $1-2-1$ lattice; the dotted lines represent the maximum and the minimum value of the theoretical velocity profile, computed with the viscosity derived by the numerical experiment: Reynolds number $R \approx 3000$, viscosity $\nu = 0.013 \pm 0.002$, typical velocity $v_* = 0.014 \pm 0.001$.

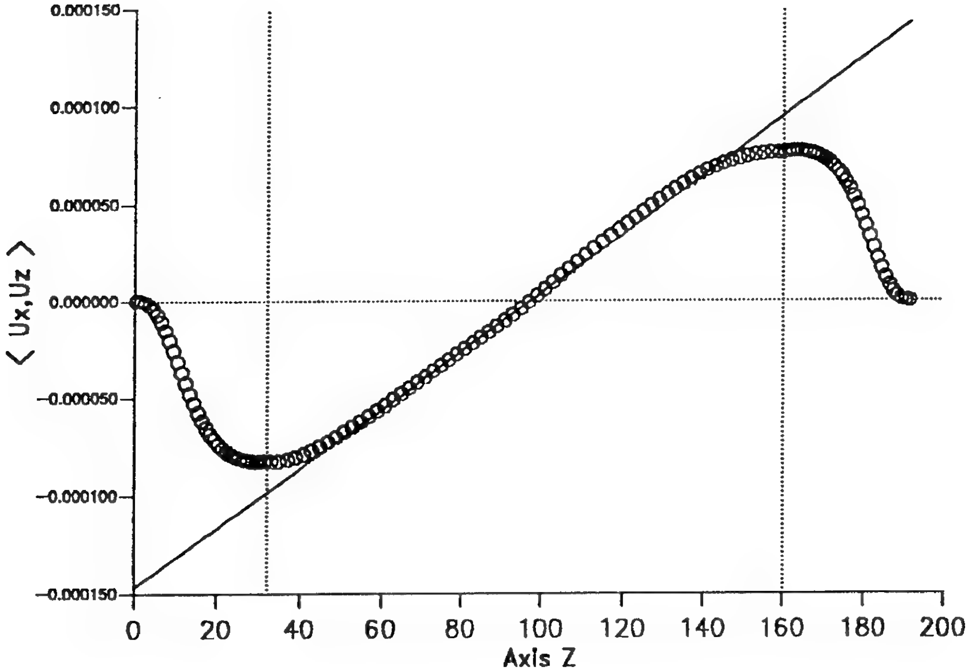


Figure 2. Stress tensor $\tau_{xz} \equiv < u_x u_z >$ vs. z, for the same conditions as Fig. 1 As a consistency check, we compute $v_* = (\tau/\rho)^{1/2}|_{z=0}$ and we obtain $v_* = 0.012$ in good agreement with the value deduced from the velocity profile.

TRANSITION IN THE CASE OF LOW FREE STREAM TURBULENCE

V.T. Grinchenko and V.S. Chelyshkov

Institute of Hydromechanics, NAS of Ukraine,
8/4, Zhelyabov st., Kiev, 252057, Ukraine

INTRODUCTION

It is well known that two types of transition are possible in the boundary layer: natural and 'bypass' transition (see review of A.M. Savill [14]). First type of transition is observed in the artificial case of low free stream turbulence, 'bypass' transition usually takes place in real technical equipment: aircraft, turbine engine *etc.* Theoretical investigations of both type of transition excite such difficult questions as problem of model construction, problems of accurate and effective space and time resolution.

Known models can be divided onto two parts: semi-empirical models (for instance, Savill-Launder-Younis model [15] (1995)) and models based on reduction of initial-value and boundary problem for Navier-Stokes equations (adding of artificial term of mass force adopted by Laurien E. & Kleiser L. [11] (1989), Parabolised Stability Equations model, which was designed by Bertolotty F.P., Herbert Th. & Spalart P.R. [1] (1992), 'fringe' model suggested by P.R. Spalart [17] (1993)). We describe now one model of second type, namely, the Slow and Fast disturbances interaction Model (SFM) designed by V.S.Chelyshkov [6] (1993). The model is based on the assumption that slow and fast disturbances interaction in longitudinal coordinate is possible in such weakly non-parallel flows as non-gradient and gradient boundary layers, jets and wakes. This idea was developed last years in the papers [4, 5, 6, 8] (see also review by V.T. Grinchenko & V.S. Chelyshkov [9]). The approach is valid for 3-D flows, but we shall regard for simplicity 2-D boundary layer near semi-infinite flat plate.

THE MODEL OF DISTURBANCE INTERACTION

It is known that two scales of flow in longitudinal coordinate (slow and fast) are possible near a flat plate. Blasius flow is slow (weekly non-parallel) flow. Two dimensional perturbances of Blasius flow are divided into two types: slow undamping perturbances, which control the boundary layer thickness [12] and fast non-stationary perturbances [16]. Both types of perturbances must depend on slow longitudinal coordinate, but experimental and theoretical investigations show, that we can neglect this dependence for

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

second ones [16]. Fast disturbances self-interaction results in fast and slow disturbances. The last ones make the contribution to weekly non-parallel flow compound. So the way of SFM construction is following. Let l be the distance from leading edge to a fixed point on a flat plate, U_∞ – a velocity of run flow, ρ – the fluid density, ν – the kinematic viscosity coefficient. Cartesian coordinates (x', y') are introduced to describe 2D non-stationary flow, which depends on time t'. These coordinates beginning coincides with leading edge, and x'-axis directs along the plate. The velocity vector components are designated as u', v' in this coordinate system, p' is the pressure. We choose the non-dimensional variables using the formulae:

$$x' = lX_0, \quad y' = \delta^\star y, \quad t' = lU_\infty^{-1}T, \quad u' = U_\infty u, \quad v' = U_\infty \lambda v,$$

$$p' = \rho U_\infty^2 p, \quad \delta^\star = \kappa\sqrt{\nu l/U_\infty}, \quad \kappa = 1.72078766, \quad \lambda = \delta^\star/l.$$

Then the velocity vector field and the pressure

$$\mathbf{F} = \{u, v, p\}(X_0, y, T)$$

is described by Navier-Stokes and continuity equations

$$\begin{gathered}\partial_T u + u\partial_{X_0}u + v\partial_y u = -\partial_{X_0}p + \frac{1}{\kappa^2}(\partial_{yy} + \lambda^2\partial_{X_0X_0})u,\\ \lambda^2(\partial_T v + u\partial_{X_0}v + v\partial_y v) = -\partial_y p + \frac{\lambda^2}{\kappa^2}(\partial_{yy} + \lambda^2\partial_{X_0X_0})v,\\ \partial_{X_0}u + \partial_y v = 0.\end{gathered} \tag{1}$$

Equations (1) need suitable initial-value and boundary conditions in the flow domain, which is not defined for the present. Boundary conditions

$$u\mid_{y=0} = v\mid_{y=0} = 0, \quad u\mid_{y\to\infty} = 1, \quad v\mid_{y\to\infty} = 0 \tag{2}$$

are set on a flat plate and far from the wall. Poisson equation for pressure

$$-(\partial_{yy} + \lambda^2\partial_{X_0X_0})p = 2\lambda^2((\partial_{X_0}u)^2 + \partial_y u\partial_{X_0}v) \tag{3}$$

is the result of equations (1).

Parameter λ is small far from the leading edge and Cartesian coordinates (X_0, y) is stretched out of transverse coordinate. When $\lambda\to 0$ Blasius solution

$$\mathbf{F} \equiv \mathbf{F}^B, \quad \mathbf{F}^B = \{u^B, v^B, p^B\}(X_0, y), \quad \mathbf{u}^B = \{u^B, v^B\},$$

satisfies Prandtl equations and the boundary conditions

$$u^B\mid_{y=0} = v^B\mid_{y=0} = 0, \quad u^B\mid_{y\to\infty} = 1, \quad v^B\mid_{y\to\infty} < \infty. \tag{4}$$

Thus physical condition of damping v when $y\to\infty$ is substituted for limitness condition.

We define

$$X_0 = 1 + X, \quad X = \lambda x, \quad T = \lambda t, \quad \mathrm{Re} = \kappa^2/\lambda. \tag{5}$$

The disturbed flow field is considered in half-band

$$\mathcal{D} = \{-\pi/\alpha \leq x \leq \pi/\alpha,\ \ 0 \leq y < \infty\},$$

where $\alpha \sim O(1)$ is a parameter. Now we define the solution $\mathbf{F}$ of the problem under consideration in the following way

$$\mathbf{F} = \mathbf{F}^B + \mathbf{F}^S + \mathbf{F}^f$$

$$\mathbf{F}^S = \{u^S, v^S, p^S\}(X_0, y, t), \quad \mathbf{u}^S = \{u^S, v^S\} \tag{6}$$

$$\mathbf{F}^f = \{u^f, v^f/\lambda, p^f\}(x, y, t), \quad \mathbf{u}^f = \{u^f, v^f\}$$

Here $\mathbf{F}^S$ and $\mathbf{F}^f$ are the vector fields describing slow and fast disturbances. We introduce the x-average in $\mathcal{D}$

$$\overline{\mathbf{F}} = \frac{\alpha}{2\pi} \int\limits_{-\pi/\alpha}^{\pi/\alpha} \mathbf{F} dx$$

and shall suppose that $\overline{\mathbf{F}}^f = 0$. Substituting (6) to initial problem (1) – (3) and throwing away, as for laminar flow description, addends of the order of $O(\lambda^2)$, a system of equations and boundary conditions are obtained. We add to nonlinear equations and subtract from them x-average of the convective addends, which contain fast disturbances. Now we can separate in the convenient way all addends of each equation into two parts. Then we break these two parts of addends and equate to zero each of them. The problem is obtained:

$$\partial_t u^S + \frac{\kappa^2}{\mathrm{Re}}(\partial_{X_0} u^B u^S + (u^B + u^S)\partial_{X_0} u^S + \partial_y u^B v^S + (v^B + v^S)\partial_y u^S) -$$

$$- \frac{1}{\mathrm{Re}}\partial_{yy} u^S + \overline{N_u(\mathbf{u}^B, \mathbf{u}^S, \mathbf{u}^f)} = 0, \tag{7}$$

$$\partial_{X_0} u^S + \partial_y v^S = 0,$$

$$u^S \mid_{y=0} = v^S \mid_{y=0} = 0, \quad u^S \mid_{y \to \infty} = 0, \quad v^S \mid_{y \to \infty} < \infty, \tag{8}$$

$$- \partial_{yy} p^S = \overline{N_p(\mathbf{u}^B, \mathbf{u}^S, \mathbf{u}^f)}, \tag{9}$$

$$p^S \mid_{y \to \infty} = \partial_y p^S \mid_{y \to \infty} = 0, \tag{10}$$

$$\partial_t u^f + N_u(\mathbf{u}^B, \mathbf{u}^S, \mathbf{u}^f) + \partial_x p^f - \frac{1}{\mathrm{Re}}\triangle u^f - \overline{N_u(\mathbf{u}^B, \mathbf{u}^S, \mathbf{u}^f)} = 0,$$

$$\partial_x u^f + \partial_y v^f = 0, \tag{11}$$

$$-\Delta p^f = N_p(\mathbf{u}^B, \mathbf{u}^S, \mathbf{u}^f) - \overline{N_p(\mathbf{u}^B, \mathbf{u}^S, \mathbf{u}^f)},$$

$$u^f \mid_{y=0} = v^f \mid_{y=0} = 0,$$

$$u^f \mid_{y\to\infty} = v^f \mid_{y\to\infty} = \partial_y p^f \mid_{y\to\infty} = 0. \qquad (12)$$

Here

$$N_u(\mathbf{u}^B, \mathbf{u}^S, \mathbf{u}^f) \equiv (u^B + u^S)\partial_x u^f + \frac{\kappa^2}{\mathrm{Re}}(\partial_{X_0} u^B + \partial_{X_0} u^S)u^f + \qquad (13)$$

$$+\frac{\kappa^2}{\mathrm{Re}}(v^B + v^S)\partial_y u^f + (\partial_y u^B + \partial_y u^S)v^f + u^f \partial_x u^f + v^f \partial_y u^f,$$

$$N_p(\mathbf{u}^B, \mathbf{u}^S, \mathbf{u}^f) \equiv 4\frac{\kappa^2}{\mathrm{Re}}(\partial_{X_0} u^B + \partial_{X_0} u^S)\partial_x u^f + \qquad (14)$$

$$+2(\partial_y u^B + \partial_y u^S)\partial_x v^f + 2((\partial_x u^f)^2 + \partial_y u^f \partial_x v^f).$$

In our opinion the SFM (7) – (14) describes near-wall flow in both cases of low and high free stream turbulence. The equations have no the second y-derivative of v^S. That is why the physical condition of damping v^S far from the wall is replaced here, like in (4), by limitness condition, and the solution of problem (7) – (14) will not be uniformly applicable. Relationship

$$\partial_t v^S \sim O(\lambda)$$

is the condition of the model validity. This relationship cannot be established *a priory*, but seems to be acceptable due to week dependence on time of $\mathbf{F}^S$. The natural conditions of disturbances damping far from the wall, like in (2), have to be carried out for "fast part" of flow field. Substitution of one of Navier-Stokes equations for Poisson equation allows us to construct time discretization schemes without the need for fractional step. This way also gives the possibility to extend the solution algorithm to 3D-problem.

The values of velocity vector components are unknown at the boundaries orthogonal to the wall. We cannot introduce periodicity conditions at these boundaries because the flow is weakly non-parallel. Following the idea of boundary layer coherent structures [2], we shall suppose that the flow is close to periodic in longitudinal direction and

$$(\mathbf{F}^S + \mathbf{F}^f) \mid_{x=-\pi/\alpha} = (\mathbf{F}^S + \mathbf{F}^f) \mid_{x=\pi/\alpha} + O(1/\mathrm{Re})$$

To vanish slight arbitrariness in these boundary conditions we shall construct the solutions, depending on both longitudinal coordinates in some special way.

SPACE AND TIME DISCRETIZATION

Direct methods are applied for discretization of the problem (7)–(14). The known forms of perturbances dependence on longitudinal coordinate are used for trial functions choice:

$$u^S = \sum_{j=0}^{N} U_j(\eta,\, t)\, X_0^{-\nu_j}, \quad v^S = \sum_{j=0}^{N} V_j(\eta,\, t)\, X_0^{-\nu_j - 1/2},$$

$$p^S = p^S(y,\, t), \quad \eta = y/\sqrt{X_0}, \tag{15}$$

$$\{u^f,\, v^f,\, p^f\}(x,\, y,\, t) = \sum_{|k|<K, k\neq 0} \{u_k,\, v_k,\, p_k\}(y,\, t)\, \exp(i\alpha k x). \tag{16}$$

In (15) $\nu_0 = 0$, and power indexes ν_j $(j > 0)$ are selected on the basis of vorticity exponential damping far from the wall, such as $\nu_1 = 1$, $\nu_2 = 1.887$, $\nu_3 = 2.867$, $\nu_4 = 3.8, \ldots$. Now we can expand first two terms in (6) into Taylor series in X, substitute the result to (7)–(14) and throw away addends of the order of $o(\lambda x)$ in (11). Using (15), (16) and expanding variable λx into Fourier series in (11) we can separate longitudinal coordinate by projection equations under consideration into two systems of test functions: X^k, $k = 0, 1, \ldots$, and $\exp(i\alpha m x)$, $m \neq 0$.

Sequences $X_0^{-\nu_j}$ and X^k are not orthogonal to each other in the interval of their changing. This leads to numerical difficulties for slow part of solution, when N is large, due to necessity to inverse matrix of Hilbert's type.

The next stage of approximation is the solution representation in coordinate orthogonal to the wall in the interval $[0, \infty)$. The asymptotics of the velocity and the pressure field coefficients of fast disturbances far from the wall have the form $exp(-\alpha k y)$ for near-wall modes, where $k > 0$ is Fourier harmonic number. Therefore in the problem class at issue for solution approximation in coordinate y it is convenient to use exponential polynomials (EP) orthogonal on semi-axis by weight of unity. Some computational and/or algorithmic advantage can present EP $\mathcal{E}_{n,k}(y) = exp(-ky)P_{n-k}^{(2k,0)}(1 - 2exp(-y))$ obtained by orthogonalization of exponential sequence in inverse order, starting from some number n [3]. Here $P_N^{(\mu,\nu)}$ are Jacobi polynomials. These polynomials are used for solution representation in coordinate orthogonal to wall, and sequence $\mathcal{E}_{n,k}$ is filled up by unity for approximation of vertical velocity vector component of slow disturbances. Final projection into phase space is carried out by Bubnov-Galerkin method, that allows one to use the 'boundary functions' [13] to satisfy the boundary conditions at the wall. For precise numerical integration Gauss quadrature formulae derived in terms of properties of EP is applied, so $3n/2$ points are used in the algorithm.

The described way of spatial approximation results in triangular matrix as discrete analog of Laplacian that allows one to employ explicit schemes in time. So variant of Runge-Kutta method was adopted for time resolution.

The following stage of discretization is stated in details in [4]. Collocation method is more preferable for 3-D flow modelling. Variant of collocation method, namely, combined direct method is suggested in [7] for near-wall flow simulation.

NUMERICAL RESULTS OF NATURAL TRANSITION SIMULATION

Level of flow vorticity far from the wall $y = 0$ and inflow boundary conditions define the influence of free stream turbulence in $\mathcal{D}$-domain. Really, recent experiments [19] show, that high free stream vorticity before a flat plate changes Blasius profile and excites fast oscillations near the nose part of the plate. So both time-undamping slow perturbances and fast disturbances are developed due to changing of inflow conditions at the boundary of $\mathcal{D}$-domain.

We shall consider here more simple case of exponentially small free stream turbulence. In this case we shall suppose that influence of time-undamping slow perturbances is small for natural transition and slow part of disturbed flow is one-dimensional in the boundary layer coordinates (η, X_0), so $N = 0$ in (15). We omit the terms of the order of $O(X)$ in equations (11), so periodicity conditions are valid at the orthogonal to the wall boundaries for fast disturbances. Such simplifications lead to initial-value and boundary problem, which has no functional arbitrariness in space. We also shall suppose that modes of continuous spectrum are not excited and our algorithm is constructed in such a way that disturbed flow damps far from the wall in accordance with asimptotics of near-wall modes.

Physical parameters Re = 520 and $\alpha = 0.308$ set 2-D flow domain. Simulation parameters are $K = 7$, $n = 32$. The parameter values yield dynamic system, which has 409 degrees of freedom. The simulation was performed for interval $0 \leq t \leq 20000$. Initial values of amplitudes were determined from the solution of Orr – Sommerfield eigenvalue problem. The values correspond to initiation of Tollmein – Schlichting wave with phase velocity equal to 0.396. The disturbance development picture is divided into the two parts. At first (t is less than ~ 10000) the travelling wave regime with increasing amplitude arises. When the oscillation energy reaches some value the single-wave regime is reconstructed and the regime close to oscillations with many frequencies is excited. Let us τ_w is disturbed skin friction and

$$\tau_w = \tau_w^S + \tau_w^f.$$

For steady flow regime the amplitudes τ_w at $x = 0$ and τ_w^S are shown in Fig. 1 and Fig. 3. Power spectrum of τ_w is shown in Fig. 2, where integers are the numbers of x-Fourier

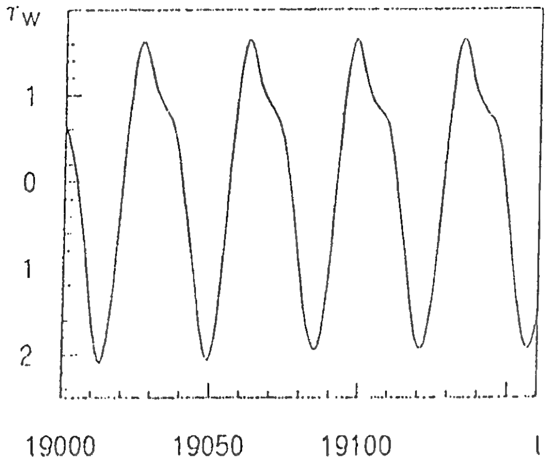


fig. 1

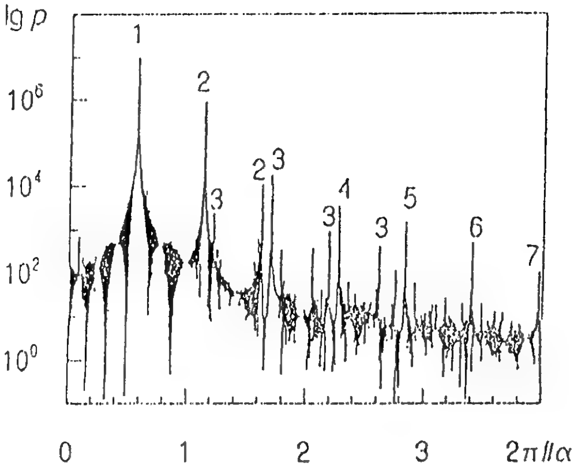


fig. 2

harmonics, which have oscillations with frequencies according to the picks. One can see that apart from main travelling wave, which has phase velocity equal to 0.566, other oscillations exist. Among these oscillations the largest energy has the oscillation with convective speed equal to $0,809\,U_\infty$, which is excited by the second x-Fourier harmonic. It is of interest that each space scale has own number of oscillation frequencies. It also appears, that near-wall travelling wave phase velocity practically coincides with near-wall propagation velocity of perturbations in channel [10]. In contrast with the result of work [13] we have found that phase velocities of both pressure and friction equal to each other near the wall. The skin friction x-Fourier harmonics $f_w(\alpha k)$ decay rate is shown in Fig. 4 for simulation time $t \sim 20000$. One can see that the decay is enough rapid, so the seventh harmonic is about 200 times less than the first.

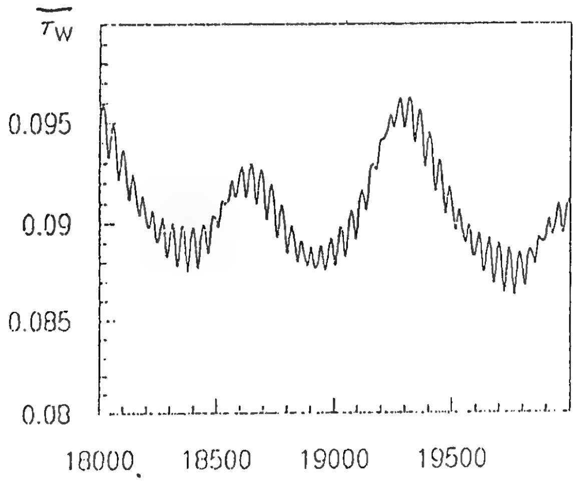


fig. 3

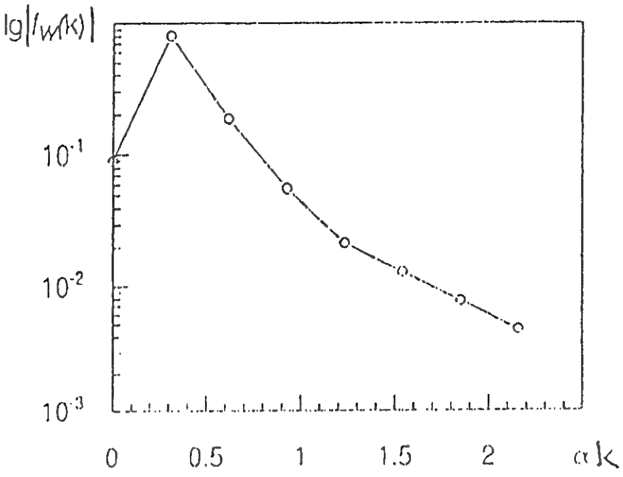


fig. 4

Direct numerical simulation experience leads to the conclusion that non-dimensional time, which is necessary to obtain fully developed flow, usually is very long. Curiously, the according physical time is enough short. Let us τ is the dimensional time, so

$$\tau = \nu \,\mathrm{Re} \,/\, U_\infty^2 \, t.$$

If $U_\infty = 1m/s$ and the fluid is water, then the physical simulation time is $10,4\,s$ in examining case. This time greatly differs from the computer time, which is necessary for 2-D modelling. Simulation of 3-D boundary layer is more difficult problem and the statistically steady solution have not been obtained up to now in this case (see, for instance, [18]).

CONCLUSION

1. The new mathematical model based on Navier-Stokes equations has been developed. The model can be effective for quantitative description of a class of weakly non-homogeneous flows. The model was tested by consideration the flow stability problem near a flat plate.

2. To verify our model approach and discretization algorithms we have carried out long-time DNS of disturbed Blasius flow for various but moderate numbers of degrees of freedom.

3. We have found that balance between the numbers of taken in orthogonal directions functions have to be observed. If K is the number of taken Fourier harmonics in longitudinal direction and n is the number of taken exponential polynomials in orthogonal to the wall direction then $n = n(K)$ for successful execution of our algorithms. It is essential to notice that increasing of K leads to n-increasing.

4. Our experience of near-wall flow modelling leads to the conclusion, that numerical solution breakdown, the so-called 'turbulence arising' does not correspond to the real physical phenomena in the boundary layer.

5. In our opinion we have found statistically steady state of flow near a flat plate. This flow is time-organized structure, which has the background of quasi-periodic oscillations with incommensurable frequencies.

References

1. Bertolotty F.P., Herbert Th. & Spalart P.R. Linear and non-linear stability of the Blasius boundary layer// J. Fluid Mech, 1992, **242**, 441.

2. Cantwell B. J. Organized motion in turbulent flows//Ann. Rev. Fluid Mech, 1981, **13**, 457.

3. Chelyshkov V.S. Sequences of orthogonal on semi-axis exponential polynomials// Doklady AN UkSSR, 1987, ser. A, No 1, 14 (in Russian).

4. Chelyshkov V.S. Numerical simulation of self-excited oscillations in the boundary layer over a flat plate. 2. Two-dimensional non-linear problem. Kiev, 1990, 20 p. (Prepr. AN UkSSR. Institut Matematiki.- 90.24) (in Russian).

5. Chelyshkov V.S. Numerical simulation of 2-D secondary flows in the boundary layer //Doklady AN UkSSR, 1990, Ser. A, No 11, 43 (in Russian).

6. Chelyshkov V.S. The model of near-wall flow//NAS of Ukraine. Institute of Hydromechanics. Annual report for 1993. Kiev, 1994, 54.

7. Chelyshkov V.S. Variant of direct method in the theory of hydrodynamic stability// Gidromekhanika, 1994, No 68, 105 (in Russian).

8. Chelyshkov V.S. Wave regime in the boundary layer// NAS of Ukraine. Institute of Hydromechanics. Annual report for 1994. Kiev, 1995, 41.

9. Grinchenko V.T., Chelyshkov V.S. Direct numerical simulation of boundary layer transition. In: Near-Wall Turbulent Flows, Ed. So, Speziale and Launder. Elsevier, 1993, 889.

10. Kim J., Hussain F. Propagation velocity of perturbations in turbulent channel flow // Phys. Fluids , A, 1993, **5**, Pt. 3, 695.

11. Laurien E. & Kleiser L. Numerical simulation of boundary-layer transition and transition control// J. Fluid Mech., 1989, **199**, 403.

12. Libby P., Fox H. Some perturbation solutions in boundary layer theory. Part 1. The momentum equation// J. Fluid Mech., 1963, **17**, No 3, 433.

13. Orszag S.A. Numerical simulation of incompressible flows within a simple boundaries. 1. Galerkin (spectral) representation// Stud. Appl. Math., 1971, **50**, 4, 293.

14. Savill A.M. Some Recent Progress in the Turbulence Modelling of Bypass Transition// In: Near Wall Turbulent Flows. Ed. So, Speziale and Launder. Elsevier, 1993, 889.

15. Savill A.M. The Savill-Launder-Younis (SLY) RST Intermittency Model for Predicting Transition// In: ERCOFTAG bulletin, 1995, No 24, 37.

16. Schlichting H. Boundary Layer Theory, 7th ed, New-York, McGrow-Hill, 1979.

17. Spalart P.R. & Watmuff J.H. Experimental and numerical study of a turbulent boundary layer with pressure gradients// J. Fluid Mech., 1993, **249**, 337.

18. Spalart P. R., Crouch J. D., Ng L. L. Numerical study of realistic perturbations in three-dimensional boundary layer. In : AGARD Symp. on application of direct and large-eddy simulation to transition and turbulence. (1994, April 18–21) Chania, Crete, Greece.

19. Westin K. J. A., Boiko A. V., Klingmann B. G. B., Kozlov V. V., Alfredsson P. H. Experiments in a boundary layer subjected to free stream turbulence. Part 1. Boundary layer structure and receptivity // J. Fluid Mech.,1994, **281**, 193.

Structured Adaptive Sub-Block Refinement for 3D Flows

K. Becker
Department EF 11
Daimler-Benz Aerospace Airbus GmbH
D-28183 Bremen

S. Rill
Hochschule Bremen
Hunefeldstr 1-5
D-28199 Bremen
Germany

1 SUMMARY

Structured sub-block refinement is a means to refine a mesh at certain areas within the flow region, in order to enhance the local resolution of the flow equations or flow solution without going to costly global mesh refinement. By the use of appropriate sensors, the regions of refinement can be defined during the running flow solving process so that the adaptation becomes automatic. And the use of structured refinement, i.e. refinement by block-like areas, does only require minor changes to the overall multi-grid iteration scheme. Strategies for the selection of sub-blocks and first results for 2D and 3D Euler- and Navier-Stokes test cases are given. The drawbacks and the potential of the method are discussed.

2 LIST OF SYMBOLS

A	continuous Navier-Stokes operator
u	continuous solution to Navier-Stokes system
f	right hand side of Navier-Stokes system
I, II	transfer operators between meshes
τ	local truncation error
I, J, K	indices of points in computational space

Subscripts and superscripts

h	discrete form referring to mesh h
$2h$	discrete form referring to mesh $2h$
$^{2h}_{h}$	from mesh h to mesh $2h$ or mesh h relative to mesh $2h$
$\tilde{\cdot}$	approximation to $\cdot$

3 INTRODUCTION

The process of discretization of the flow equations causes differences between the continuous solution of the Navier-Stokes system of differential equations and the solution of the system put onto the computer. This error is called local truncation error, and it plays the major role concerning solution deficiencies. Discretization errors, their magnitude and distribution about the flow region, are influenced by geometrical mesh properties as well as properties of the flow solution. Both types have to be encountered when selecting appropriate sensors that shall drive adaptative flow solving algorithms.

All types or combinations of sensors result in single pointwise quantities which have to be scanned. A certain threshold determines whether a point or local region is already o.k. with respect to the expected error or whether it is a candidate for mesh adaptation.

In principle, mesh adaptation distinguishes between mesh enrichment and mesh movement. Mesh movement tries to improve the solution by shifting the existing mesh points to more appropriate positions. Mesh enrichment means to refine the mesh which leads to an increased number of mesh points. Eventually, a coarsening of the mesh is also possible in regions where the quality measure is already good. Both approaches have their specific problems, best may be to combine them.

Within this paper, we try to describe adaptive mesh enrichment strategies within a structured multi-block context. The principle structure of the flow solver shall not be affected by the local refinement. This means that refinement zones have to be of structured type, i.e. they must be regular mesh blocks. We use a concept of sub-blocks which has been developed within the Euromesh project of BRITE/EURAM and the ECARP project of IMT Aera 3 of CEC research. This concept allows to treat sub-blocks as additional levels of refinement in the usual multi-grid sequence of the MELINA flow solver (Fig. 1) [RilBec92].

Structured mesh enrichment of the form described above has its drawbacks when tracing features of the flow which run diagonally through the mesh. Unless there are limiters for the size of the sub-blocks, quite large refinement zones must be expected. So, we are aware that this special type of mesh enrichment will not be the ultimate but a first and practicable solution to mesh adaptation.

4 SUB-BLOCK APPROACH

The idea of structured sub-block refinement is to simply patch locally refined mesh blocks onto the existing mesh and connect the additional fine sub-blocks with the original mesh via a multigrid technique. Thereby, a sub-block has to lie completely in a grid block of the existing mesh, which includes touching the block boundary. But a grid block may have various sub-blocks and a sub-block may have several sub-blocks itself (see Fig. 2).

The sub-block approach can be viewed as a compromize between structured an unstructured meshes, combining the benefit of high computational efficiency on structured

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

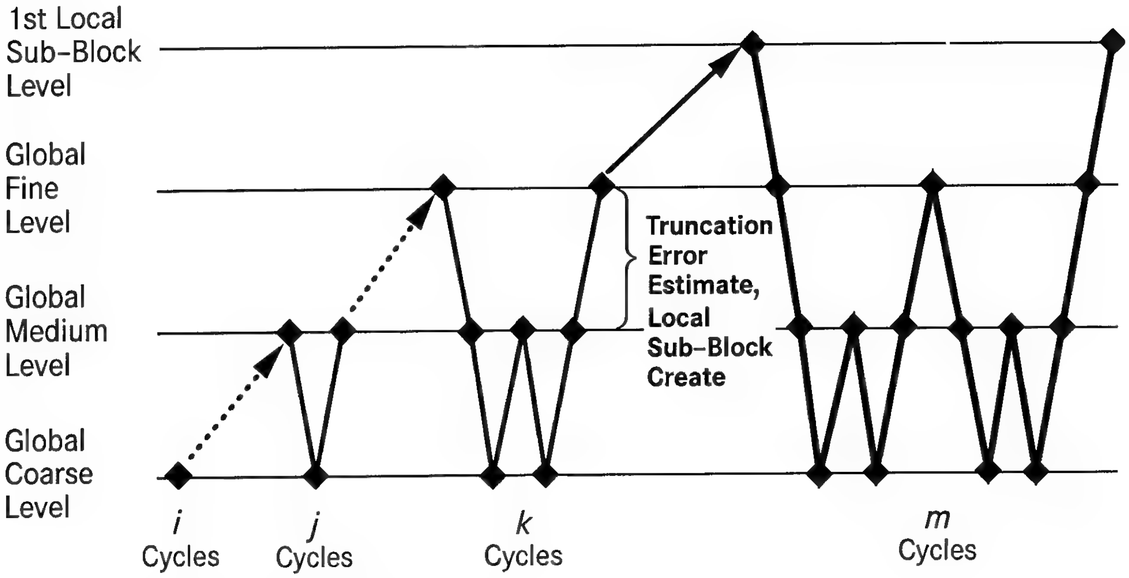


Figure 1: Multigrid sequence with local refinement.

meshes and of clustering grid points in a "quasi unstructured" way by scattering sub-blocks and even further refined blocks in regions of discretization errors. It is envisaged to use this method for solution adaptive mesh refinement if the regions of sub-block refinement are determined automatically during the iteration by suitable sensor functions.

4.1 Surface and Interior Point Definition

When a sub-block is created, between each two mesh points on a coarse grid line an intermediate fine grid point has to be introduced.

On any component's surface, this new point has to lie on the surface. This means that the new point has to be constructed using the original surface definition. However, this causes severe problems if the surfaces are defined by external CAD means, for example. Therefore, most often special interpolation procedures are used which create local surface approximations from the existing coarse mesh points. The single approaches differ by the quality of surface representation. For aerodynamics, the criteria of absolut distances to the real CAD surface and wavyness of the interpolated surface play the major role. For the moment, we don't want to stress this problem: we simply use Coons' local patches.

The definition of interior fine mesh points is not that constrained. As long as Euler meshes are considered, those mesh points can be constructed using simple trilinear interpolation of the coarse cells in the field.

For the very dense Navier-Stokes meshes, in the vicinity of a curved surface intersections of field mesh lines with the true boundary are very likely to occur with trilinear interpolation. Therefore the filling algorithm has been changed to the use of Coons' representation for each mesh plane parallel to the surface, not only the surface planes. This guarantees smooth behaviour of the mesh in the whole sub-block, especially in the boundary layer mesh. Additionally, it avoids any intersection of mesh lines or planes with fixed surfaces. Because this approach is that robust, fast and easy, we adopted it also for the Euler meshes.

4.2 Communication between Sub-blocks and Coarse Blocks

In general, sub-blocks cover only part of the computational domain. Boundary conditions on their outer block boundaries must be defined such that there is no algorithmic influence on the overall flow solution. Within the multigrid context, flow variables are interpolated from the coarse mesh. If the sub-block boundary touches the coarse block boundary, the same boundary condition is applied. Wall, symmetry or similar conditions are thus treated correctly. Special things have to be done if the sub-block boundary lies inside the coarse block. Boundary values of the sub-block cannot be set as fixed Dirichlet type conditions because this conflicts with the mixed type nature of the flow equations. The interpolated values serve only as initial guess and the values are updated using the original flow equations themselves on the fine mesh. Therefor at least one row of guard cells has to be created around the sub-block which contains the flux integral information needed for the application of the cell vertex discretization at the real sub-block boundary. This procedure is quite the same as is applied between two adjacent blocks of the original non-refined mesh. In addition to this, co servativeness has to be ensured across the sub-block boundaries. In our code, this is achieved by replacing the flux integrals along coarse cell faces at the sub-block boundary location: the coarse mesh integral is replaced by the sum of the participating fine mesh integrals.

This type of communication between sub-blocks and blocks is managed with the help of the face group concept. Each block has at least one face group. This group of six faces consists of the minimum/maximum index planes (boundaries of the computational domain) of the block. For each sub-block that is added to the coarse block, a new face group is defined. It contains those

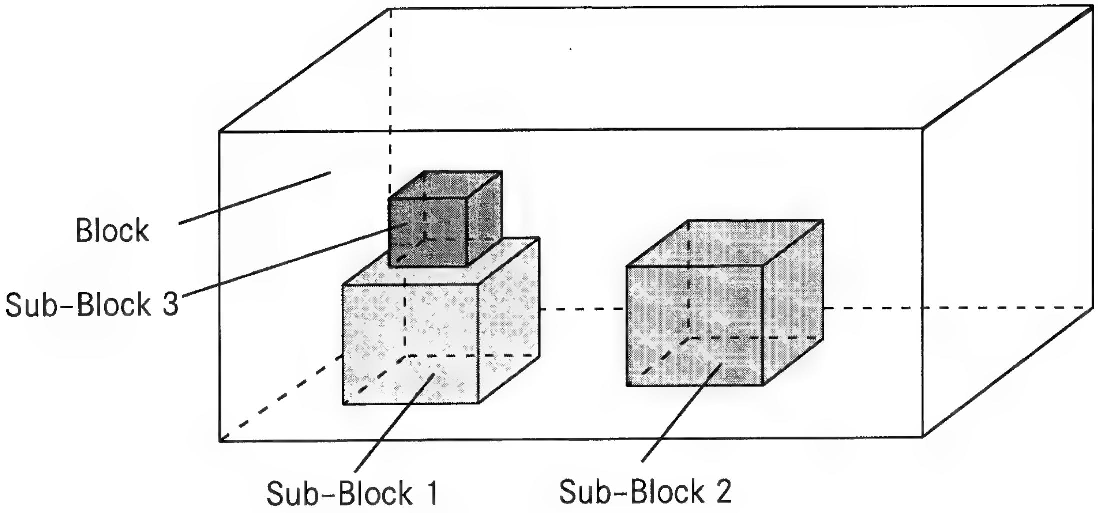


Figure 2: Sub-blocks within a mesh block - schematic view.

segments of coarse mesh planes that coincide with the block boundaries of the respective local sub-block. So this face group is the hull of the sub-block inside the coarse block. The respective topological description data are used to drive the communication of flow variables and other relevant data within the flow solver.

If two sub-blocks within a coarse block or across the boundaries of two coarse blocks are adjacent to each other, then communication should be allowed directly between those sub-blocks. The simplest way is to transfer data from a sub-block to the respective face group of the coarse block and from there to the neighbouring sub-block. However, this path of communication contains interpolation errors and should thus be replaced by the immediate transfer of data from one sub-block to its neighbour. Within the topological description data, this problem could be easily solved because sub-blocks are treated in the same way as usual blocks.

If a new sub-block is constructed, the topological data are updated automatically. Boundary conditions and connections to adjacent sub-blocks are detected and included in the description. This makes the fully adaptive incorporation of new sub-blocks into an existing multi-block mesh relatively easy once the respective coarse mesh face group boundaries are known. One major technical difficulty is the generality of sub-block to sub-block connections. Up to now, two sub-blocks of a coarse block are only allowed to touch each other if it is with one full face. Touching only with part of a face would require new segmentation of the respective faces and can easily result in very complex face segmentations. On the other hand, the above restriction hinders an effective treatment of diagonal refinement. For the moment the drawback of full face touching has to be overcome by resizing respective sub-blocks. However, part-of-face touching is under development.

Several topologically different sub-block configurations have been tested. Because the sensor evaluator may suggest quite general addition of sub-blocks, it might be necessary to have such arrangements run quite robust. For example, if we have a four block finest mesh, in a first adaptation step sub-blocks might be suggested only for three blocks. This leads to different finest levels on different blocks within the multi-grid cycles. Additionally, consecutive sub-blocking during subsequent adaptation loops has to be allowed which means that sub-sub-sub-...blocks can occur. Such and similar conditions have been investigated concerning the convergence behaviour and the quality of solution, especially at the junction of refined and non-refined regions. No specific problem has been detected with the Euler flow solver. However, with the Navier-Stokes solver it turned out that the implementation of the turbulence model has a great impact. In practice, the Baldwin-Lomax model used requires wall distance information. This information is very difficult to obtain in general multi-block meshes if it is not evaluated in a preprocessing step.

5 SENSOR EVALUATION

The evaluation of any sensor field always means scanning the field for a pre-specified range of values that are considered to indicate deficiencies of solution accuracy. We can distinguish between sensors that depend on the flow solution itself and sensors that are defined by purely geometrical quantities. Mathematical analysis of discretization leads to certain guidelines concerning the mesh. One of those rules is that one should use smooth and orthogonal meshes. Measures of those quantities can thus be used to determine "bad" regions within an existing grid. On the other hand, the flow itself shall drive the mesh in order to properly resolve special features like shocks, stagnation regions, boundary layers or shear layers. The analysis of respective sensors leads to suggestions for enhanced grid density regions.

5.1 Flow Independent Sensors

Within the BRITE/EURAM Euromesh project, a palette of geometrical quality measu: es has been developed. Now we use these measures for a priori qualification of meshes,

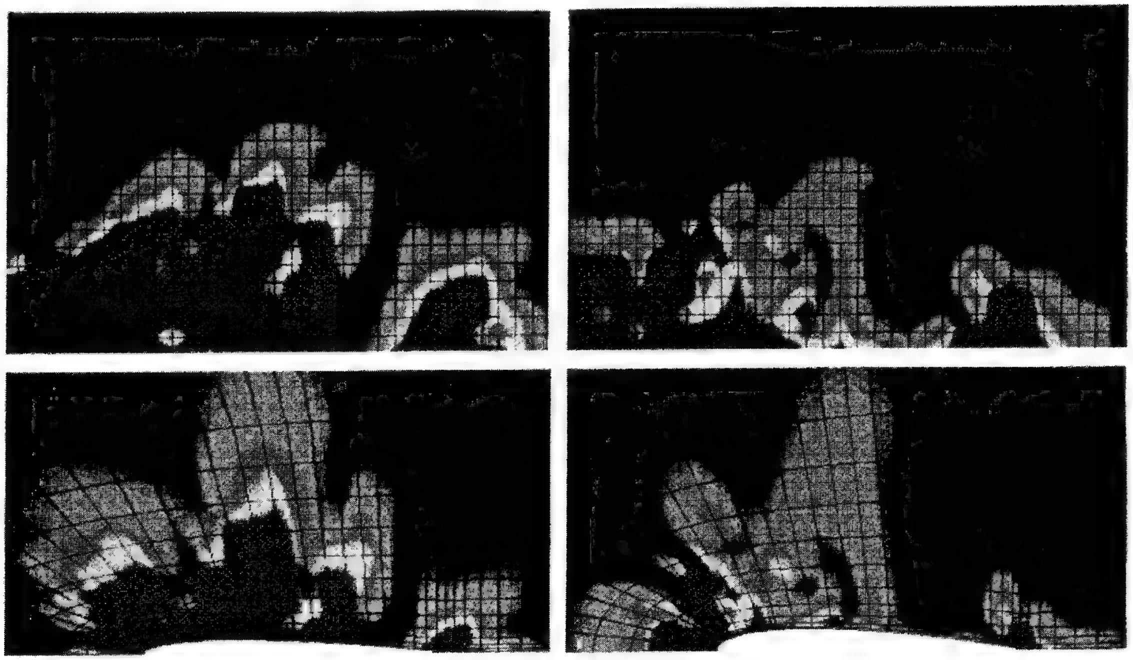


Figure 3: Local truncation error estimate for a NACA0012 Euler case - computational space and physical space - left: τ(*continuity equation*), right: τ(*2nd momentum equation*).

mainly. Within the DA mesh generation system INGRID, the following 3D measures are implemented: orthogonality, skewness, aspect ratio and expansion rate for 3 index directions. Neither of them leads to an absolute criteria for mesh quality. Orthogonality, for example, cannot be achieved in the whole mesh if there appear angles other than 90 degrees on the surface. Respectation of those angles is necessary for high quality surface representation, but clearly violates the principle of orthogonality. Similar statements can be made for the other quantities. Nevertheless, those quantities should be taken into account when creating base meshes.

5.2 Flow Dependent Sensors

The first and most likely reason for deficiencies in solution accuracy is a too high level of local truncation error. This error describes in principle how good the nonlinear operators of the Navier-Stokes equations are approximated by the discrete differentiation and integration rules on a specific mesh. It must be reminded that there is no locality in the relation between this error and the global truncation error of the solution itself, i.e. the solution error can occur at quite different locations than the local truncation error [Klim95]. This is especially due to the transport character of the equations.

Truncation error estimates can be extracted directly from the multi-grid cycles: Specific differences between medium and coarse mesh residuals in a three level computation yield an estimate of the local truncation error τ [Bra77]. This estimate for all equations of the Euler or Navier-Stokes system is used to define the locally refined (fine) mesh level.

In detail, if the continuous equation

$$Au = f \qquad (1)$$

is discretised on a mesh with typical mesh size h

$$A_h u_h = f_h \qquad (2)$$

where u_h is the discrete solution, then the local truncation error τ_h is defined by

$$\tau_h = A_h u - Au \qquad (3)$$

If we further add and subtract the discrete operator A_h applied to an approximation $\tilde{u}_h$of the discrete solution u_h,

$$A_h u_h = f_h - A_h\tilde{u}_h + A_h\tilde{u}_h, \qquad (4)$$

and represent this equation on the next coarser grid with mesh size $2h$, then we end up with the multigrid coarse grid correction equation

$$A_{2h}u_{2h} = II_h^{2h}(f_h - A_h\tilde{u}_h) + A_{2h}I_h^{2h}\tilde{u}_h, \qquad (5)$$

which contains the local truncation error estimate on mesh $2h$ relative to mesh h:

$$\tau_h^{2h} = A_{2h}I_h^{2h}\tilde{u}_h - II_h^{2h}A_h\tilde{u}_h. \qquad (6)$$

Under the assumptions $A_h \approx A$ and $\tilde{u}_h \approx u$ this yields

$$\tau_h^{2h} \approx A_{2h}u - Au, \qquad (7)$$

which is the local truncation error τ_{2h} on mesh $2h$. Fig. 3 gives an impression on the distribution and the levels of local truncation error for a 2D transonic test case.

During the studies it has been found very useful to have presentations of the estimate in physical as well as in

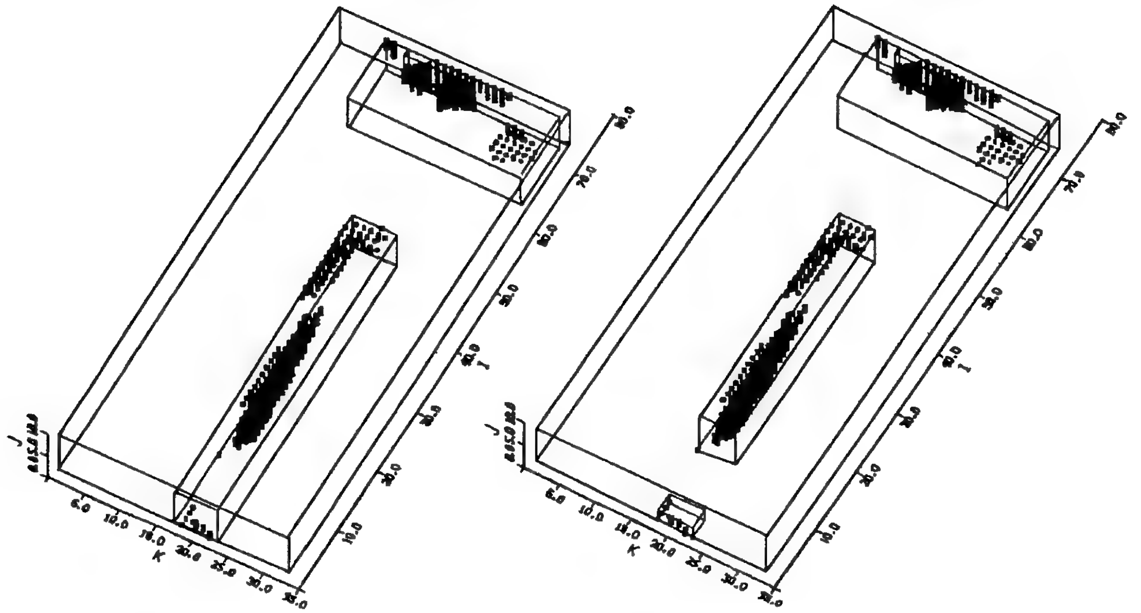


Figure 4: Suggestions for new sub-blocks - left: RADIUS=5, right: RADIUS=2.

computational domain. Interestingly, the errors for the single equations seem to be complementary to each other. Near the nose of the airfoil, $\tau(continuity - equation)$ suggests refinement in other parts of the flow field than $\tau(momentum - equations)$, for example. For our investigations, the L1-norm over all equations was taken to drive the refinement. More detailed studies can be found in [Lau95].

5.3 Sub-block Definition

In the context of structured sub-block refinement, a strategy has to be developed by which the location and extension of local sub-blocks can be determined. The evaluation of any sensor defines a set of "bad" points or cells that appear as clouds in the index space of each structured block. On the one hand, the sub-blocks have to cover those clouds. On the other hand, the size of the sub-blocks corresponds to the numerical effort and thus has to be as small as possible. The strategy to define reasonable sub-blocks is as follows:

- Find a first bad point (I,J,K).
- Set IMIN=IMAX=I-index of bad point; same with J and K indices.
- Trace the surroundings (IMIN-RADIUS, IMAX+RADIUS; ...) of the current (IMIN,IMAX; JMIN,JMAX; KMIN,KMAX) area for more bad points.
- If any more bad point has been identified, enlarge the respective MIN/MAX values and restart search.
- If no more bad point can be found, define the sub-block from the current MIN/MAX values.

The user-given tolerance value RADIUS has a strong influence on the size of the sub-blocks. It also defines the minimum distance between two sub-blocks within one block. In order to avoid that very large sub-blocks are suggested which more look like a global mesh refinement the maximum size of sub-blocks must be bounded. Also, it may happen that many small sub-blocks are created if any singular bad point is taken into account. This can be hindered by a minimum bound for the number of points within a sub-block.

Fig. 4 shows the index cube representation of suggestions for sub-blocks within coarse block. In the first case, a RADIUS of 5 was chosen whereas in the second case the RADIUS value was 2, resulting in one more sub-block of smaller size.

5.4 Adaptation Cycle

Mesh enrichment via sub-blocks should run automatically within the flow solution process. However, for the development of such a method it is reasonable to combine the single elements of code in a more loose form. The adaptation cycle has been splitted into 4 steps:

- Start calculation on a reasonably fine mesh and store the results (mesh, flow solution, local truncation error),
- Run the sub-block suggestion code and store MIN/MAX indices for each coarse block,
- Generate the enriched mesh which contains the previous mesh and the new sub-blocks,
- Restart the flow solver using interpolated values as starting solution for the new sub-blocks.

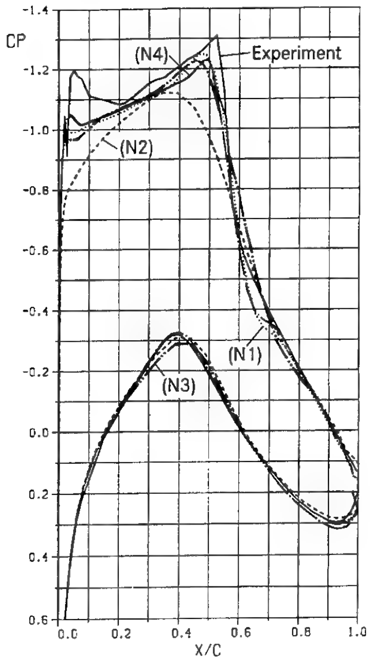


Figure 5: Pressure distribution RAE2822 for meshes (N1),..,(N4) - comparison with experiment.

This cycle can be run until the maximum number of refinement levels has been reached. It is assumed that each time only the relatively finest level can be refined.

6 NUMERICAL RESULTS

The sub-block concept described above has been implemented in 3D. However, for cost reasons and for first validation purposes it is reasonable to begin with 2D Euler and Navier-Stokes flows. The basis of 3D Euler and Navier-Stokes investigations on local refinement and adaptation was a wing/body combination.

6.1 2D Test Cases

Local mesh refinement has been tested in 2D, first: RAE2822 test case 9 with a free stream Mach number of 0.734, angle of attack of 2.54 degrees and Reynolds number of 6.5 million. The Navier-Stokes calculation should serve as a preliminary test to show the effect and effectiveness of local refinement. Refinement was done by hand using sub-blocks which covered the whole upper surface including the supersonic region and extended slightly on the lower surface near the nose of the airfoil. Fig. 5 shows the resulting pressure distributions for different meshes and the experimental values. We have chosen four different meshes as there were

(N1) standard fine C-mesh with 241 × 77 mesh points, about 30 points normal to the wall in the boundary layer,

(N2) mesh (N1) coarsened once by omitting every second point, with 121 × 39 mesh points,

(N3) mesh (N2) with sub-blocks and

(N4) mesh (N1) with sub-blocks.

The RAE2822 test case 9 has often been used for validation purposes. Always problems with the suction peak on the upper wing nose have been reported as it is the case with the present results. Current computations have been made for fully turbulent flow.

If (N1) is assumed to be a mesh of usual fineness, the (N1) result should be the target for adaptation. Results produced with the coarser mesh (N2) obviously show up high level numerical errors. If (N2) is refined locally as described above, which is (N3), the result is already very close to the target (N1). However, the computing time is only about 40 p.c. of the (N1) computation, as can be seen from Fig. 6. Additional local refinement for (N1), which is (N4), yields again a solution which is more close to the experiment both near the nose and for the pressure gradient in front of the shock. For cost reasons, a target computation for a globally refined (N1) mesh has not been performed. The experiment has been used, instead. In the (N4) case, convergence of the lift coefficient is reached at only minor additional expense compared to (N1).

Fig. 6 shows the convergence behaviour of the method for the different meshes. The residuals for all cases drop down with CPU time very quickly. There are no specific observations in the case of embedded sub-blocks being present. However, the current implementation of the Baldwin Lomax turbulence model in the MELINA flow solver may cause problems if the sub-block cuts the mesh within the boundary layer. If such a sub-block does not extend down to the wall surface, then the wall distance needed for the turbulence model has not the right values and may thus lead to bad results or even non-convergence of the overall algorithm. This state of the flow solver hinders automatic adaptation in any complex case at the moment.

6.2 3D Wing/Body Test Case

The application of the sensor analysis implemented in the ADAPTOR code [LauMau95] to the F4 wing/body Navier-Stokes test case showed up nice properties. As can be seen from Figs. 7,8, with the current base mesh the τ-error is concentrated in the vicinity of the configuration. It clearly detects

- the body nose region as being not properly resolved,
- the wing nose region as spurious entropy production region,
- the shock region as being insufficiently resolved for steep gradients,
- the sonic line as being sensitive to numerical errors,
- the trailing edge and wake region as being sensitive because of rapidly changing flow including free shear layers and
- the boundary layer near the wall where pre-adaptation of the mesh to the boundary layer profiles is only possible up to a certain extent.

This makes us hope that automatic recognition of deficiencies in discretization is possible, and adaptation will reduce the overall local truncation error.

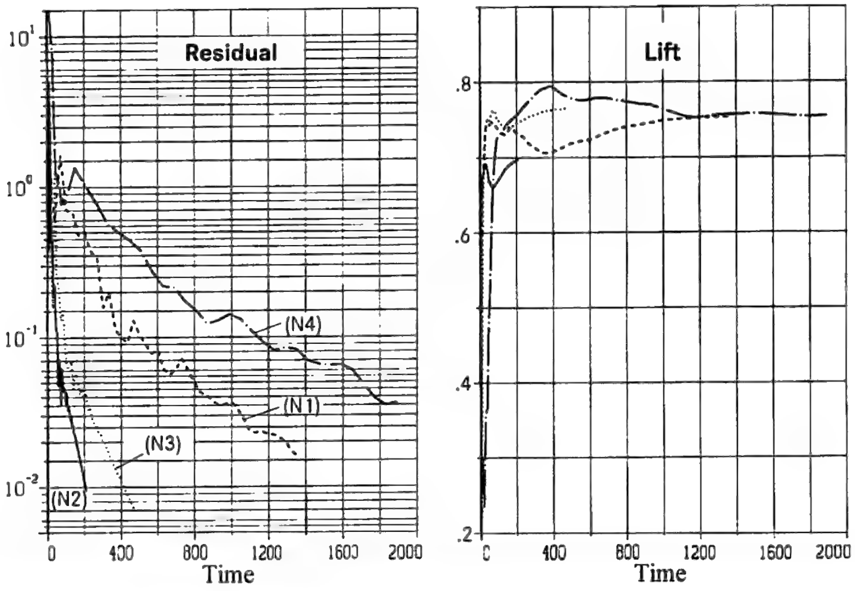


Figure 6: Convergence behaviour for meshes (N1),..,(N4) - residual and lift coefficient against CPU time.

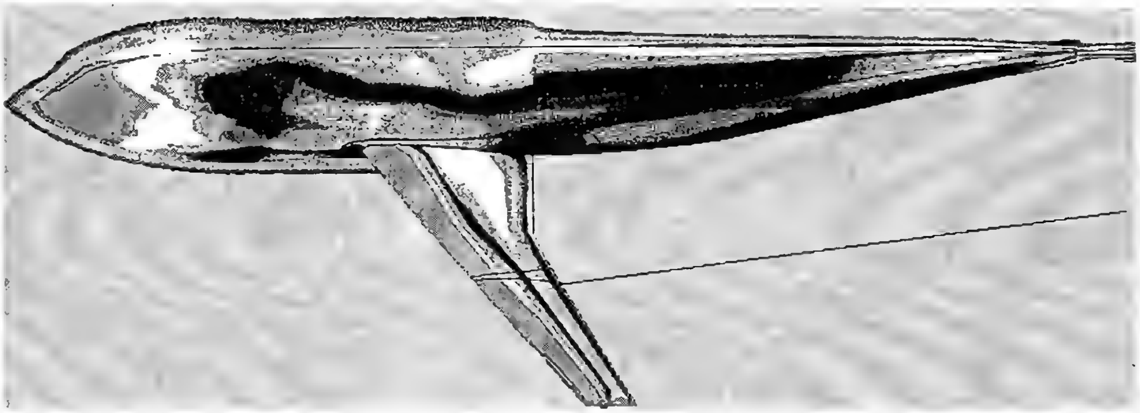


Figure 7: F4 wing/body configuration - continuity equation truncation error estimate for surface and symmetry plane.

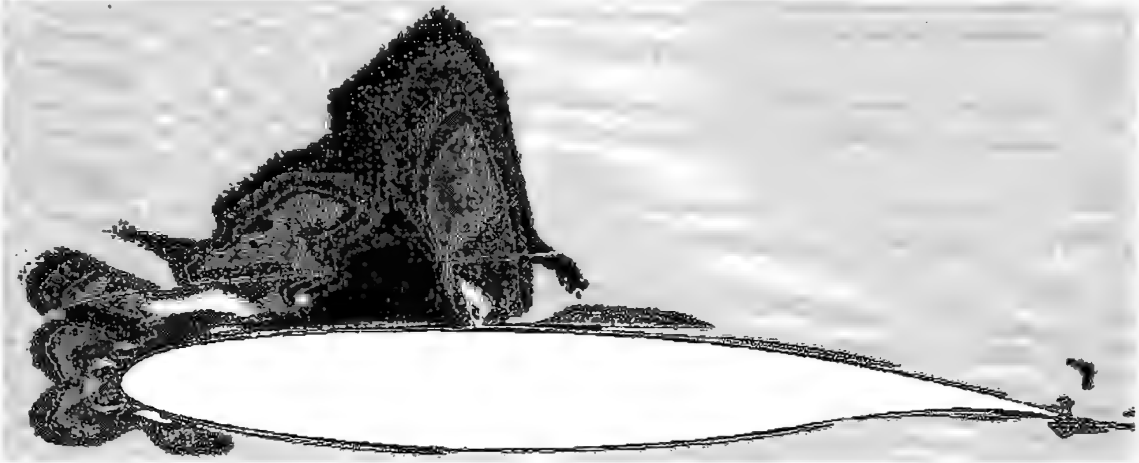


Figure 8: F4 wing/body configuration - truncation error estimate of x-momentum equation for spanwise mesh plane.

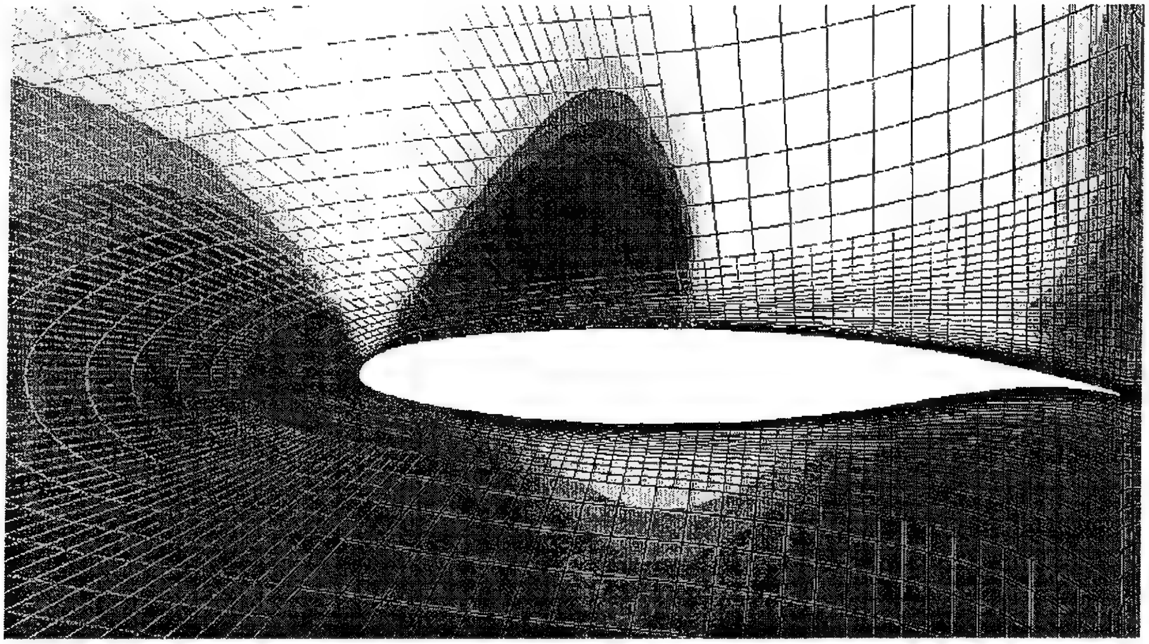


Figure 9: F4 wing/body configuration - Mach contours and sub-blocks at spanwise mesh plane.

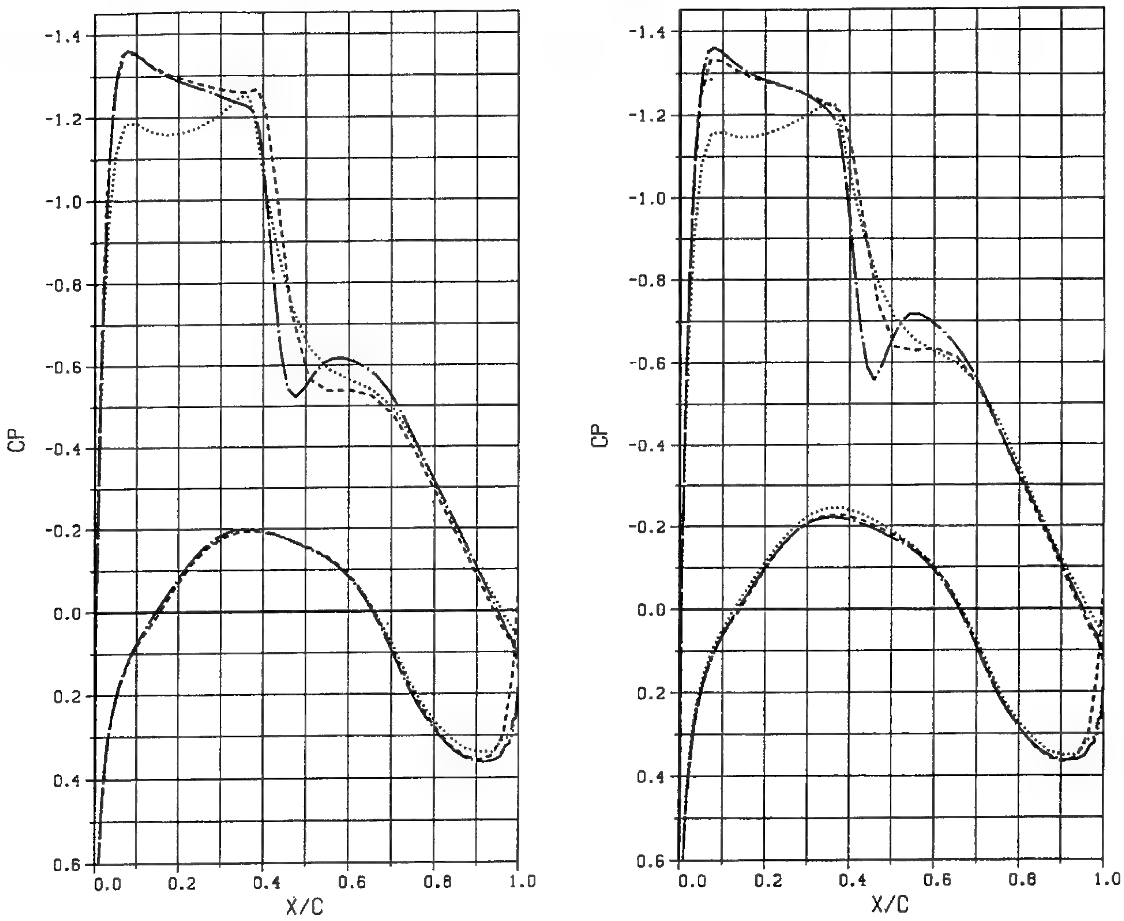


Figure 10: F4 wing/body configuration - cp at 63 p.c. (left) and 52 p.c. (right) wing span ; –.– fine reference ; ...... non-refined ; - - - with embedding.

The ADAPTOR run with the medium grid truncation error estimate of all 5 flow equations leads to multiple sub-blocks. It mainly suggests a block in the vicinity of the surface along the whole span of the wing, starting somewhere below and behind the wing and extending over the upper surface again behind the trailing edge. A second block covers the off-surface region around the wing nose and extends about the supersonic region. Parts of the sub-blocks can be seen in Fig. 9, where a spanwise cut with local Mach contours is shown. The pressure distribution at two mid-wing cuts show quite a good improvement compared to the coarse mesh solution (Fig. 10). The suction peak as well as the pressure roof top gradient and the shock position are in a good agreement compared to the fine mesh reference solution. And the locally refined mesh has only about 40 p.c. of the points of the global fine mesh.

7 CONCLUSIONS

Mesh enrichment based on a structured sub-block approach has been considered as an effective way to improve numerical solution of flow equations. Tools have been defined and strategic provisions have been made to test this approach under industrial constraints. Up to now, the main procedures have been set up. Results for locally refined meshes have been calculated for 2D and 3D Euler- and Navier-Stokes test cases. Next step will be the full integration of the adaptation into the flow solver and the validation and improvement of the overall process.

Because of the problems with turbulence model implementation in Navier-Stokes we'll first try to sort out the automatic adaptation problems, sensor analysis, etc. on the basis of the Euler equations. More general sub-block - to - sub-block connections are under development which allow a more cost-effective resolution of diagonal flow features.

A lot of tests have been run with the ADAPTOR code, and a lot of changes of the evaluating strategy have been necessary in order to find a reasonable suggestion for sub-blocks. The expense of more than 50 p.c. cost saving which we have achieved with the current examples is already quite good under industrial conditions. Ongoing work will be concentrated on making adaptation fully automatic, robust and efficient.

8 ACKNOWLEDGEMENTS

The basis of this work has been partly conducted as BRITE/EURAM area 5, CEC funded, applied research. Recent results have been obtained within the IMT area3, CEC funded, ECARP project. We are grateful for this support.

References

[Bec93] Becker. K., Aumann, P.: "The Interactive Grid Generation System INGRID - Version 5.0", DA-report, December 1993.

[Bra77] Brandt, A.: "Multi-Level Adaptive Solutions to Boundary Value Problems", Mathematics of Computation, Vol. 31, No. 138, pp. 333-390, April 1977.

[Klim95] Klimetzek, F.: "Fehlerästimatoren für Strömungsberechnungsverfahren, Teil I (Literaturauswertung, 1993, 93-038) und II (Anwendung und Beurteilung, 1994, 94-091), Technical Report, Daimler-Benz-AG.

[Lau95] Lauke Th.: "Adaption von Rechennetzen zur Steigerung der Genauigkeit von 3D-Strömungssimulationen", Diplomarbeit, Technical University of Berlin, June 1995.

[LauMau95] Lauke Th., Mauch. H.: "ADAPTOR User's Manual", Daimler-Benz Aerospace Airbus, June 1995.

[RilBec92] Rill, S.; Becker. K.: "MELINA - A Multi-Block. Multi-Grid 3D Euler Code with Local Sub-Block Technique for Local Mesh Refinement", ICAS Paper 92-4.3.R, ICAS Conf., Beijing, Sept. 1992.

Multiblock Structured Grid Algorithms for Euler Solvers in a Parallel Computing Framework

Stefano Sibilla
Aermacchi S.p.A.
Dipartimento di Aerodinamica
Via Foresio, 1 21040 Venegono Superiore (VA) Italy

and

Marcello Vitaletti
IBM Semea S.p.A.
E.C.S.E.C.
Piazza G. Pastore,6 00144 Roma Italy

SUMMARY

Specific algorithms have been developed for numerical solution of Euler equations on multiblock structured grids of general topology; these algorithms involve determination of convective and dissipative fluxes, residual collection from fine grid levels during multigrid cycles and time step evaluation. They must be properly integrated with residual and flow variable averaging when the internal boundary condition is introduced.
The influence of block subdivision on the bow-shock in front of a blunt-nosed body is analysed with different multiblock algorithms; a structured and a locally unstructured topology are also compared.
Results show that no additional error is introduced in multiblock solutions if internal block boundary conditions are applied at each stage and edge/corner boundary cell contributions to flow quantities are properly taken in account.

LIST OF SYMBOLS

a	speed of sound
Cd	drag coefficient
CFL	Courant number
Cp_{st}	stagnation pressure coefficient
D	dissipative flux
E	specific energy
H	specific enthalpy
p	pressure
Q	convective flux
q	flow quantity vector
R	residual
$\bar{s}$	cell face area vector
u,v,w	velocity components
V	cell volume
V_{CV}	control volume
x,y,z	Cartesian coordinates
γ	specific heat ratio
Δt	time step
ϵ	numerical viscosity coefficient
Λ	spectral radius
ν	pressure sensor
ξ,η,ζ	curvilinear coordinates
ρ	density

1. INTRODUCTION

Multiblock methods consist in the decomposition of complex computational domains into simpler subdomains, which can be more easily handled in the management of the simulation and in the subdivision of the computational task on different processors.
Structured grid blocks can be generated in these subdomains, in order to combine the efficiency and simplicity of CFD algorithms developed for single-block structured grids with the geometric flexibility needed to describe topologically complex regions.
The main difficulty in multiblock methods lies in the correct treatment of block interfaces, which are located in the flow region and represent a numerical boundary condition with no reference to the physical problem: their presence can introduce errors in the solution which can either prevent complete convergence to the exact solution or impose constraints on the grid generation.
IBM has developed a parallel multiblock framework called PARAGRID [1,2] which supports suitable data structure for the management of data communication between adjacent blocks.
The computation is performed in parallel mode at block level, thus allowing exploitation of

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

workstation clusters and/or multi-processor systems.

A structured multiblock Euler solver had been previously implemented within this framework [3]; its results were generally good as long as the overall solution quality and global aerodynamic coefficient evaluation were concerned. Problems were nevertheless detected in the convergence rate and in the solution quality at the interfaces between adjacent blocks; moreover, only "structured" block topologies were solved consistently with original structured algorithm: this means, for example, that only internal edges shared by four blocks or corners shared by eight blocks were allowed; for all other block topologies, approximate corrections were introduced.

Some solution algorithms were therefore modified in order to account for the presence of locally unstructured topologies at block boundaries; these algorithms were designed for application in a parallel environment, minimizing the number of data exchanges between adjacent blocks, and therefore the communications between computational nodes.

2. NUMERICAL SCHEME

2.1 Finite volume formulation

The three-dimensional Euler equations

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x}f(q) + \frac{\partial}{\partial y}g(q) + \frac{\partial}{\partial z}h(q) = 0 \quad (1)$$

where

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix} \; ; \; f = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ \rho u H \end{bmatrix} \; ;$$

$$g = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho v w \\ \rho v H \end{bmatrix} \; ; \; h = \begin{bmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ \rho w H \end{bmatrix} \; , \quad (2)$$

are written in integral form

$$\frac{\partial}{\partial t}\iiint_V q \cdot dV + \iint_{\delta V} (f \cdot ds_x + g \cdot ds_y + h \cdot ds_z) = 0 \quad (3)$$

and are solved through a cell-vertex finite volume space discretization [4]: flow quantity values located at cell corners represent average values of flow quantities in the *control volume* made of all the cells (e.g. 8 for an internal node of a structured grid) sharing that node.

Convective fluxes through the control volume surface, which are represented by the second term in the left hand side of (3), are computed as sum of the contributions of all the cell faces which form the control volume surface itself; face values are taken as the average of the values at the corners of the face.

Such scheme is equivalent to a second-order accurate central difference on a Cartesian grid; such discretization leads to odd-even decoupling, allowing numerical oscillations, and provides no intrinsic numerical dissipation to damp these oscillations and other non-linear instabilities. A dissipative term, based on first- and third-order differences of the flux variables and scaled on the local spectral radii of the flux Jacobians, is introduced in the form of an added flux term [5]. For a control volume centered on the grid point *i,j,k* equation (3) takes the semi-discretized form

$$\frac{d}{dt}(V_{CV}\, q_{I,J,K}) + Q_{I,J,K} + D_{I,J,K} = 0 \quad . \quad (4)$$

In equation (4) $Q_{i,j,k}$ is the discretized convective flux

$$Q_{i,j,k} = \sum_{n=1}^{NF_{i,j,k}} (f_n \cdot s_{x_n} + g_n \cdot s_{y_n} + h_n \cdot s_{z_n}) \quad (5)$$

where $NF_{i,j,k}$ is the number of cell faces forming the surface of the control volume centered on node *i,j,k* and having area vector $\bar{s}$; the form of the dissipative flux $D_{i,j,k}$ is discussed in section 4. Equation (4) is solved in time by a 5-stage Runge-Kutta scheme [6] whose coefficients are chosen in order to allow high stability limits (CFL=4 on the linear convection equation) and large margins for numerical dissipation. Stability limits can be increased by two or three times if residuals are smoothed by application of a suitable implicit operator at the end of each intermediate Runge-Kutta stage. Finally, multigrid method for the reduction of low frequency errors [4] accelerates convergence to steady state.

2.2. Domain decomposition

The computational domain is divided into smaller hexahedral structured blocks; each of the

faces of a block is either part of a physical boundary or it is an interface to an adjacent block. An enlarged computational block is built, adding to the original core a two-layer halo extending in all the blocks sharing a boundary face, edge or corner with the original subdomain. Updated flow values are available in the halo regions when data exchange is performed.
Although made up of structured parts, the enlarged subdomain can show locally unstructured regions at core edges or corners (figure 1).
Equation (5) for the determination of the convective flux depends only on the determination of the number *NF* of cell faces that form the control volume surface; it can be applied straightforwardly to structured as well as to unstructured topologies. If updated values are available in all the enlarged copmpuational blocks, identical values of Q_K are computed for the different replicas of node *K*.

2.3. Local time step computation

Local time step must be computed from available data in each stuctured and unstructured control volume in a minimum number of computational steps in order to reduce the number of data exchanges. At the end of each step, updated flowfield quantities are available only in the core region of the block. Cell spectral radii are computed as sum of contributions in each grid-coordinate direction: for ξ-direction one obtains

$$\lambda_\xi = |u\cdot\xi_x + v\cdot\xi_y + w\cdot\xi_z| + a\cdot\sqrt{\xi_x^2+\xi_y^2+\xi_z^2} \tag{6}$$

and similar expressions for η- and ζ-directions, which sum up into the local spectral radius

$$\Lambda = \lambda_\xi + \lambda_\eta + \lambda_\zeta \quad . \tag{7}$$

To minimize data exchange needs in the computation of spectral radii at block interfaces, the product of cell contributions (6) and of cell volumes

$$\pi_\xi = \lambda_\xi \cdot V = \vec{v}\cdot\vec{S} + \sqrt{\gamma\frac{p}{\rho}}\cdot|\vec{S}| \tag{8}$$

is computed in each core region, and

$$\Pi = \Lambda \cdot V = \pi_\xi + \pi_\eta + \pi_\zeta \tag{9}$$

is derived instead of (7). Data exchange of Π values is performed at this point): having built Π as a cell quantity instead of a nodal one, no averaging step is required and computational overhead is minimum.
Being the time step in the control volume relative to a node

$$\Delta t = \frac{CFL}{\Lambda_{CV}} = \frac{CFL}{\sum_{n=1}^{NC}\Lambda_n} \tag{10}$$

where *NC* is the number of cells which build up the control volume *CV*, a convenient modified time step is obtained from (9):

$$\begin{aligned}\tilde{\Delta} &= \frac{CFL}{\Pi_{CV}} = \frac{CFL}{\sum_{n=1}^{NC}\Pi_n} = \\ &= \frac{CFL}{\bar{\Lambda}_{CV}\cdot V_{CV}} = \frac{\Delta t}{V_{CV}} \quad ,\end{aligned} \tag{11}$$

where the average spectral radius in the control volume

$$\bar{\Lambda}_{CV} = \frac{1}{V_{CV}}\cdot\sum_{n=1}^{NC}\lambda_n V_n \tag{12}$$

has been introduced, and Π values are available in the whole extended domain.
Modified time step (11) is directly introduced into the time-discretized form of (4)

$$\begin{aligned}q_{I,J,K}^{n+1} &= q_{I,J,K}^{n} + \frac{\Delta t_{I,J,K}}{V_{CV}}(Q_{I,J,K}^{n} + D_{I,J,K}^{n}) = \\ &= q_{i,j,k} + \tilde{\Delta}_{i,j,k}(Q_{i,j,k}^{n} + D_{i,j,k}^{n})\end{aligned} \tag{13}$$

from which updated values of flow quantities are obtained.

2.4. Multigrid residual driving

In multigrid methods, the residual of numerical solution of (4) on the fine grid is used to "drive" the residual evaluation on the coarser ones, i.e. coarse grid steps are used to determine corrections to fine grid residuals rather than completely new residual values.

A simple algorithm has been used to collect fine grid nodal residuals to coarse grid nodal "driver residuals" with small computational effort and validity on structured as well as unstructured topological entities.

A fine grid cell residual is computed as

$$C^{(F)}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \sum_{\xi=i}^{i+1}\sum_{\eta=j}^{j+1}\sum_{\zeta=k}^{k+1} \frac{R^{(F)}_{\xi,\eta,\zeta}}{n_{\xi,\eta,\zeta}} \qquad (14)$$

where $n_{i,j,k}$ is the number of cells sharing fine grid node *i,j,k* (figure 2-a). After cell data exchange, coarse grid nodal "driver residual" values are obtained by sum of the contributions of the fine grid cells which share the coarse grid node:

$$R^{(C)}_{I,J,K} = \sum_{n=1}^{NC_{I,J,K}} C^{(F)}_{n} \quad . \qquad (15)$$

Fine grid cell values (14) contribute to one coarse grid node only and the algorithm guarantees correct evaluation of driver residuals on unstructured nodes automatically (figure 2-b).

3. BLOCK INTERFACE CONDITION

3.1. Data exchange strategies

Contiguous blocks share nodes on boundary faces and/or edges and/or vertices, while cells belong to a single block only; in a cell-vertex formulation, where flow variables are defined at nodes, different values may be computed in replicas of the same boundary node owned by different blocks. The PARAGRID framework ensures that the same average value is assigned to all such replicas of a boundary node at the end of each block update step, when data exchange between blocks is performed.

Three different implementations of the multigrid algorithm have been studied. In the first implementation the block update step includes a *full multi-grid cycle*, with frozen halo data. In the second implementation the block update step includes the *five-stage Runge-Kutta cycle* on a grid level and the restriction/prolongation of solution and residuals to the successive grid level. In the third implementation the block update step only includes a *single Runge-Kutta stage* on the current grid level.

The first strategy has minimum memory requirements and maximum parallel efficiency but leads to an inconsistency in the computation of the flow field at internal boundaries. In this case, the averaging process is applied to the *flow quantities* associated with all replicas of a boundary node.

In principle, the third strategy ensures identity of values assigned to different replicas of a boundary node at the price of larger memory requirements and overheads due to the more frequent exchange of halo data. In practice, small discrepancies in boundary node replicas still occur, due to the implicit nature of the residual smoothing phase which is confined to work at the block level. With this choice the averaging process is applied to the *residuals* rather than the flow quantities.

The second choice represents a compromise between the previous two: halo flow values are still frozen during the time integration, but more frequent data exchange between blocks reduces strongly the generation of interface errors; on the other hand, solution is faster and requires less CPU memory than the exact solution.

3.2. Numerical experiments

Numerical experiments show that numerical errors introduced by the first interface condition reduce local stability margins and put severe restrictions on the block subdivision of the grid: block interfaces falling in the middle of strong gradient regions can often lead to divergence of the computation.

A simple geometry, consisting in a cylindrical body ending with a spherical cap of unit radius, has been chosen to investigate the limits of the examined strategies.

Figure 3 shows different topologies used in the analysis of the blunt-nosed body at a Mach number of 2 and zero incidence. The block interface in grid "A" crosses intentionally the bow shock close to the symmetry axis, where shock intensity is higher; in the grid "B" the division surface has been moved upstream.

Single-block solutions are compared with multiblock solutions obtained by application of different interface treatments; all computations have been run for 100 multigrid steps with 3-level W-cycle, after 50+50 initialization steps on two coarser grid levels. They have all been performed in single precision.

Single-block computations (figure 4-a) show a bow-shock located in front of the nose, at a

distance of approx. 0.35 nose radii; maximum pressure coefficient at stagnation is Cp_{st} =1.63 and drag coefficient is Cd =0.7756 with reference to the cross section area.
Figure 4-b shows the flow pattern resulting from all the converged computations on the 4-block grid "A": the block interface passes at (x=-1.35, y=0), i.e. where the bow-shock intensity is maximum, but the position of the bow shock is identical, Cp_{st} =1.63 and Cd =0.7756.
Flow variable exchange and averaging at the end of the multigrid cycle (figure 5) leads to divergence on the 4-block grid "A" at a CFL number of 8 and forces either a reduction of CFL number to 6 or the adoption of the modified grid "B". Figure 6 shows that, at CFL=8, data exchange at the end of each Runge-Kutta cycle leads to convergence in 50 % more steps than exchange at each intermediate stage. Figure 7 shows that single and multiblock computations are equivalent in the latter case, and that the interface boundary condition becomes completely transparent to the computation.
A simulation of the transonic vortical flow around a wing-body-canard sharp leading edge configuration, at a Mach number of 0.85 and an incidence of 10°, has been obtained from a multiblock computation with data exchange at each Runge-Kutta stage, and compared with single-block results [7].
Pressure plots in the cross flow (figure 8-a) and on the wing surface (figure 8-b) at 0.6 wing chords show that block decomposition has slight influence on position or intensity of the vortices, although block interfaces cross both the wing leading edge and the canard vortex.

4. NUMERICAL DISSIPATION

The dissipative flux $D_{i,j,k}$ in equation (4) is based on a background term, dependent on the third order difference of the flow variables scaled on the local spectral radii of the flux Jacobians. A sensor based on the local pressure gradient switches a first order difference term in presence of flow discontinuities.
On a structured grid the dissipative flux is

$$D_{i,j,k}(q) = d_{i-\frac{1}{2},j,k} - d_{i+\frac{1}{2},j,k} + d_{i,j-\frac{1}{2},k} - d_{i,j+\frac{1}{2},k} + d_{i,j,k-\frac{1}{2}} - d_{i,j,k+\frac{1}{2}} \tag{16}$$

and each mixed first- and third-order difference term [4,6] based on local curvilinear coordinate system is

$$d_{i-\frac{1}{2},j,k} = \epsilon^{(2)} \lambda_{i-\frac{1}{2},j,k}(q_{i,j,k} - q_{i-1,j,k}) - \epsilon^{(4)} \lambda_{i-\frac{1}{2},j,k}(q_{i+1,j,k} - 3q_{i,j,k} + 3q_{i-1,j,k} - q_{i-2,j,k}) \tag{17}$$

scaled on local spectral radii components (6); numerical viscosity coefficients are based on the pressure sensor

$$\nu_{i,j,k} = \left| \frac{p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k}}{p_{i+1,j,k} + 2p_{i,j,k} + p_{i-1,j,k}} \right| \qquad \bar{\nu}_{i+\frac{1}{2},j,k} = \max(\nu_{i,j,k}, \nu_{i+1,j,k}) \tag{18}$$

and take the form

$$\epsilon^{(2)}_{i+\frac{1}{2},j,k} = V^{(2)} \min\left(\frac{1}{4}, \bar{\nu}_{i+\frac{1}{2},j,k}\right) \qquad \epsilon^{(4)}_{i+\frac{1}{2},j,k} = \max\left(0, V^{(4)} - \epsilon^{(2)}_{i+\frac{1}{2},j,k}\right) \tag{19}$$

The above formulation cannot be consistently applied to the nodes of block edges and corners where the block topology is locally unstructured: the dissipation computed for different replicas of such boundary nodes on the basis of equation (16) would span different sets of neighbouring nodes, thus leading to an inconsistency.

An unstructured formulation derived from the work of Mavriplis [8] has been tested to evaluate improvements in the analysis of flows in these regions.
An approximation to the Laplacian at the boundary node $K=(i,j,k)$ is constructed as

$$dq_K = \sum_{J=1}^{n} (q_J - q_K) = \sum_{J=1}^{n} q_J - n q_K \tag{20}$$

where the summation in (20) is performed over all the n nodes connected by a cell edge to node K.
In this case the dissipative flux becomes the sum of a Laplacian and a biharmonic operator

$$D_{i,j,k} = D_K = \sum_{J=1}^{n} (dq_J - dq_K)\,\frac{\epsilon_J^{(4)}\Lambda_J + \epsilon_K^{(4)}\Lambda_K}{2} + \sum_{J=1}^{n} (q_J - q_K)\,\frac{\epsilon_J^{(2)}\Lambda_J\nu_J + \epsilon_K^{(2)}\Lambda_K\nu_K}{2} \qquad (21)$$

where Λ is the spectral radius in the control volume relative to node $K=(i,j,k)$, the pressure sensor is

$$\nu_K = \frac{\sum_{J=1}^{n} (p_J - p_K)}{\sum_{J=1}^{n} (p_J + p_K)} \qquad (22)$$

and the nodal numerical viscosity coefficients, as in (19), are

$$\epsilon^{(2)} = V^{(2)} \min\left(\frac{1}{4}, \nu\right)$$
$$\epsilon^{(4)} = \max(0, V^{(4)} - \epsilon^{(2)}) \qquad (23)$$

Coefficients $V^{(2)}$ and $V^{(4)}$ can be set in both formulations to obtain desired properties of convergence and damping. Optimal convergence was obtained in this case with values $V^{(2)}=1$ and $V^{(4)}=1/32$ in the structured formulation, $V^{(2)}=1/2$ and $V^{(4)}=15/1024$ in the unstructured formulation.

A different grid around the blunt-nosed body has been generated: a 7-block grid showing an unstructured edge, shared by five neighbouring blocks, in the vicinity of the bow shock wave. Figure 9 shows the block decomposition and the flow pattern at Mach 2 and zero incidence; a slight deflection of the shock wave is present at the unstructured edge, but it should be ascribed to the unsuitable cell distribution in the zone.
Computations have been carried out with both dissipation schemes; plots of the logarithm of density residual in the unstructured edge nodes are compared in figure 10, showing that errors due to inconsistent computation of dissipative fluxes (17) prevent from complete convergence, even if variable averaging is performed at each Runge-Kutta stage; unstructured formulation (21) leads to complete, although slower, convergence.

5. TIME AND MEMORY REQUIREMENTS

CPU time requirements have been measured by serial runs of the blunt-nosed body test case on an IBM Risc 6000 550. These measure, together with data relative to memory occupation, is obviously dependent on the code FLO67P-2 [9] and on the PARAGRID framework: they are presented here mostly as qualitative comparison between the multiblock algorithms previously discussed.
Table 1 shows CPU times and RAM occupation needed by for the various proposed strategies; times are expressed as CPU seconds per node per iteration, memory occupation is expressed as Mbytes per thousand nodes.
Exchange at multigrid level needs 20 % less time than exchange at each Runge-Kutta stage; this partly compensates for the reduction in CFL number. On the other hand, it needs less than 50 % memory. Memory occupation can in all cases be reduced by .08 Mbyte/knode if metric coefficients are recomputed at the beginning of each block update step, at the price of higher time requirements.

6. CONCLUSIONS

The determination of the most convenient multiblock solution strategy among the examined algorithms is not immediate.
Numerical experiments of section 3.2 and 4 show that, if data exchange is performed at each intermediate stage, interface condition has no impact on stability limits and convergence rate; other conditions generate a reduction of convergence rate and, in the case of exchange only at the end of the multigrid cycle, of stability limits.
On the other hand, approximate solutions at block interfaces yield a reduction in time and mostly in memory needs.
Exchanging halo data at Runge-Kutta level is a compromise solution which retains stability bounds of the exact description with reduced time and memory requirements at the price of a slower convergence.

REFERENCES

1. Poggi F., Dellagiacoma F., Paoletti S., Vitaletti M., "Multidomain Computations of Compressible Flows in a Parallel Scheduling Environment" in *Parallel Computational Fluid Dynamics '92*, ed. R.B. Pelz,

A. Ecer and J. Hauser, pp. 111-122, North-Holland, 1992.

2. Paoletti S., Poggi F., Vitaletti M., "Paragrid - a Parallel Multiblock Environment for Computational Fluid Dynamics", proc. of *VEC/PAR '93 - 1st International Meeting on Vector and Parallel Processing*, Porto, 1993.

3. Dellagiacoma F., Vitaletti M., Jameson A., Martinelli L., Sibilla S., Visintini L., "Flo67p: a Multi-block Version of Flo67 Running within Paragrid", in *Parallel Computational Fluid Dynamics: New Trends and Advances*, ed. A.Ecer et al., pp. 199-206, North-Holland, 1995.

4. Jameson A., "A Vertex Based Multigrid Algorithm for Three Dimensional Compressible Flow Calculations", *ASME Symposium on Numerical Methods for Compressible Flows*, Annaheim, 1986.

5. Jameson A., "Transonic Flow Calculations", Princeton University MAE Report 1651, 1984.

6. Jameson A., Schmidt W., Turkel E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes", AIAA Paper 81-1259, 1981.

7. Malfa E., "IEPG - TA15 Results of Aermacchi Euler Code Around the Wing-Body-Canard WB(C)-1 Configuration in Transonic Flow Condition", Aermacchi Report 275-TA15-021, 1991.

8. Mavriplis D.J., "Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes", AIAA Journal, Vol. 28, No. 2, February 1990, pp. 213-221.

9. Sibilla S., Vitaletti M., "Cell-Vertex Multigrid Solvers in the PARAGRID Framework", proc. of *Parallel Computational Fluid Dynamics '95*, Pasedena, 1995.

data exchange at each	numerical dissipation	CPU time *s pnts*$^{-1}$ *it*$^{-1}$	RAM *Mbyte kpnts*$^{-1}$
Multigrid cycle	structured	4.47 10^{-4}	0.195
Runge-Kutta cycle	structured	4.75 10^{-4}	0.276
Runge-Kutta stage	structured	5.55 10^{-4}	0.404
Runge-Kutta stage	unstructured	5.19 10^{-4}	0.462

Table 1 Time and memory requirements for the examined multiblock algorithms.

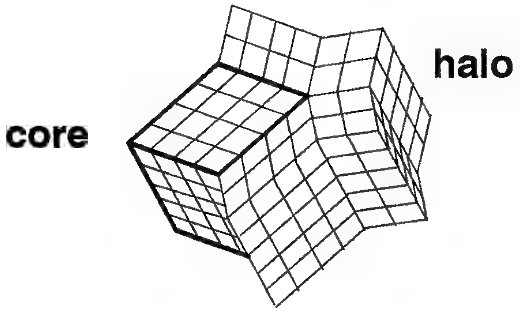


Figure 1 Example of unstructured local topology in a multiblock structured grid: edge shared by five blocks.

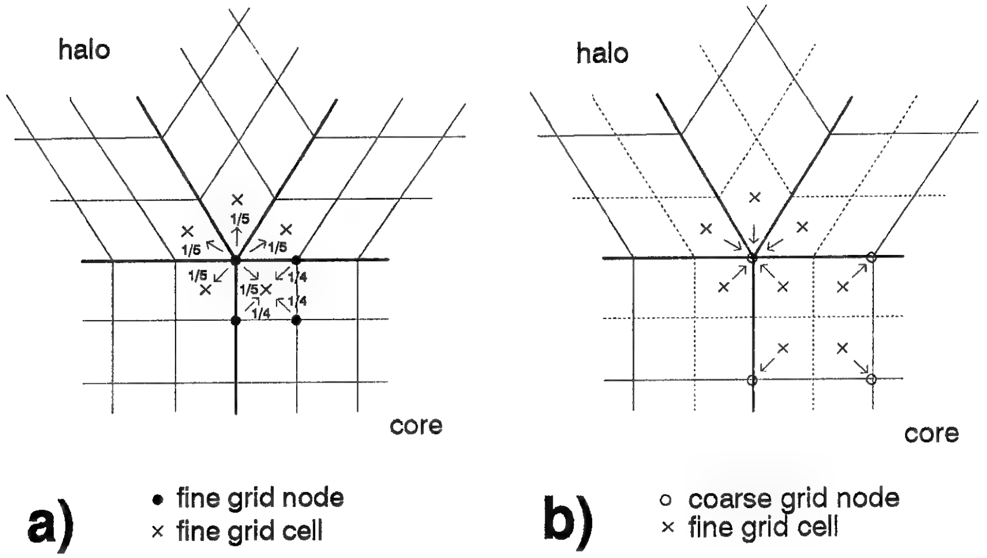


Figure 2 Algorithm for multigrid "driver residual" computation: a) contribution of fine grid nodes to fine grid cell values (14); b) contribution of fine grid cell values to coarse grid nodal "driver residuals".

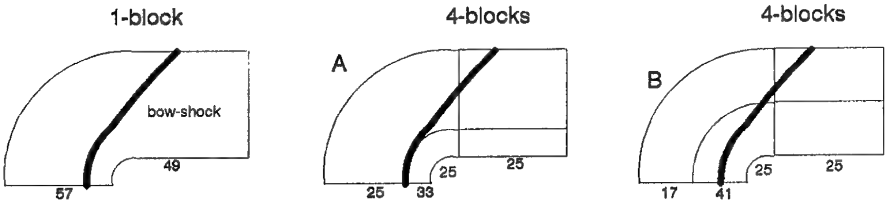


Figure 3 Blunt nosed body flow at Mach 2: Flow features and grid topologies.

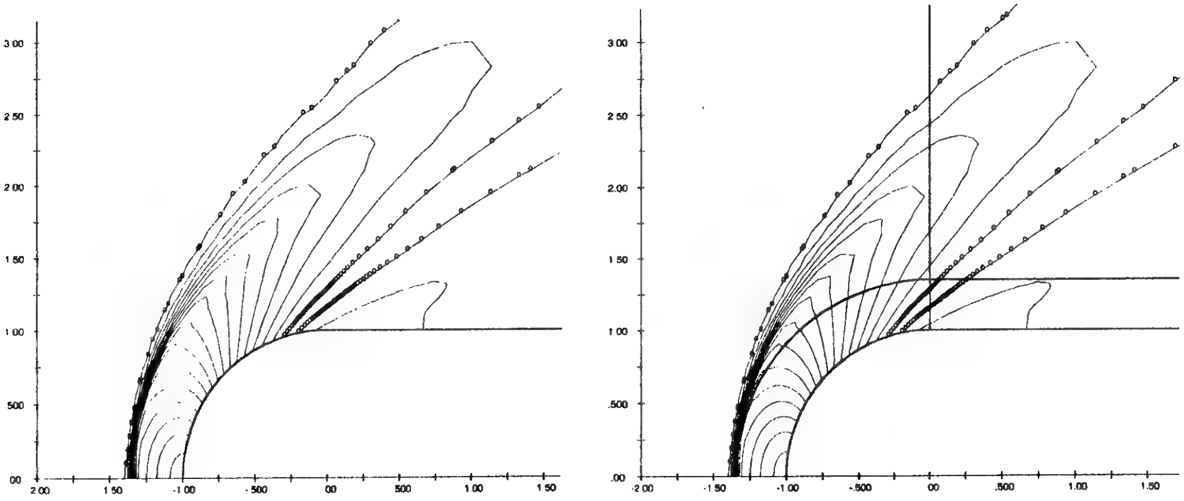


Figure 4 Iso-pressure plot of the flow around a blunt nosed body (Mach$=$2, $\alpha=0$):a) single-block computation; b) multiblock computation on grid "A".

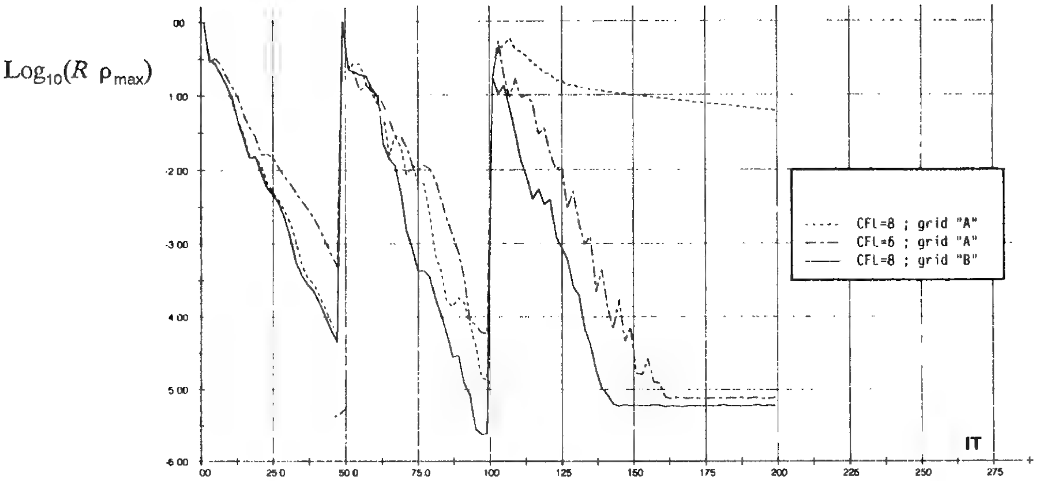


Figure 5 Convergence history of maximum density residual for multiblock solution of the flow around a blunt nosed body (Mach=2, $\alpha=0$):data exchange performed at the end of each multigrid cycle.

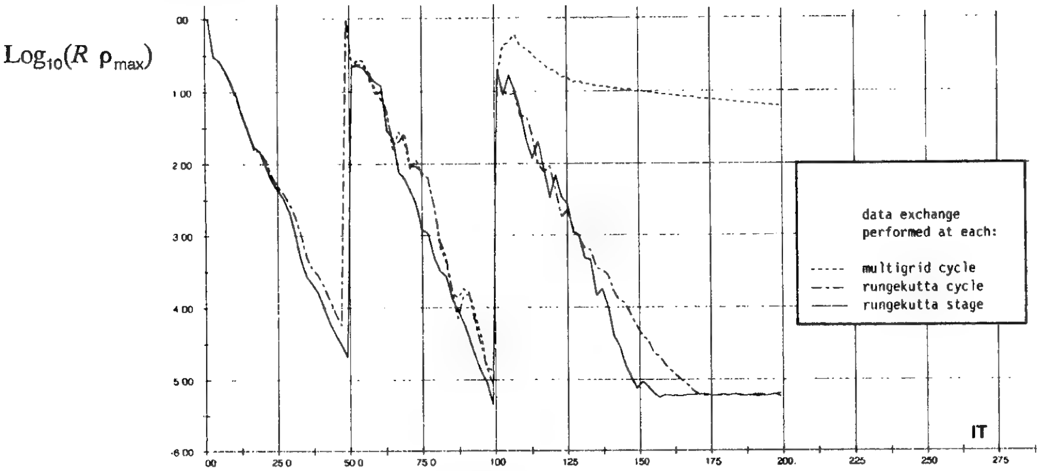


Figure 6 Convergence history of maximum density residual for multiblock solution of the flow around a blunt nosed body (Mach=2, $\alpha=0$):behaviour of different strategies for data exchange between grid blocks at CFL=8.

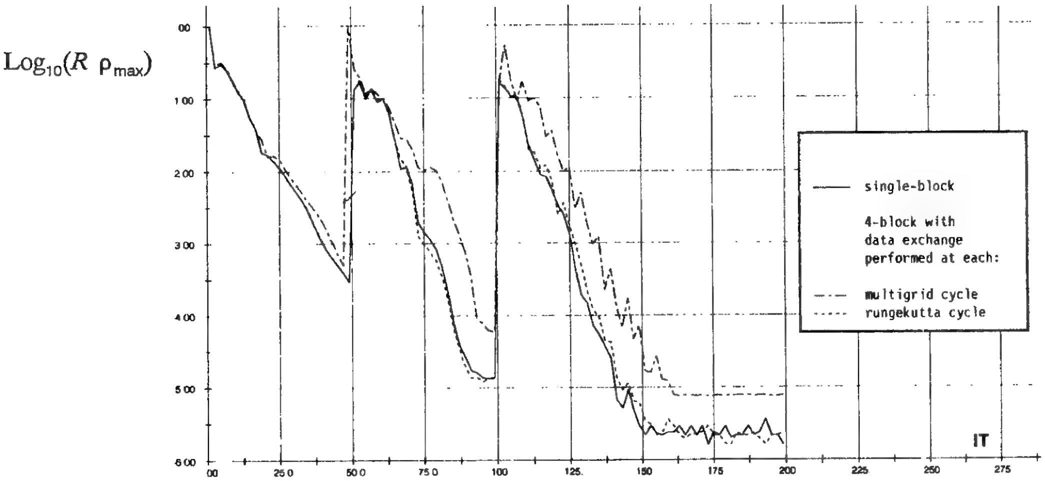


Figure 7 Convergence history of maximum density residual for multiblock solution of the flow around a blunt nosed body (Mach=2, $\alpha=0$): comparison between single-block and multiblock computations at CFL=6.

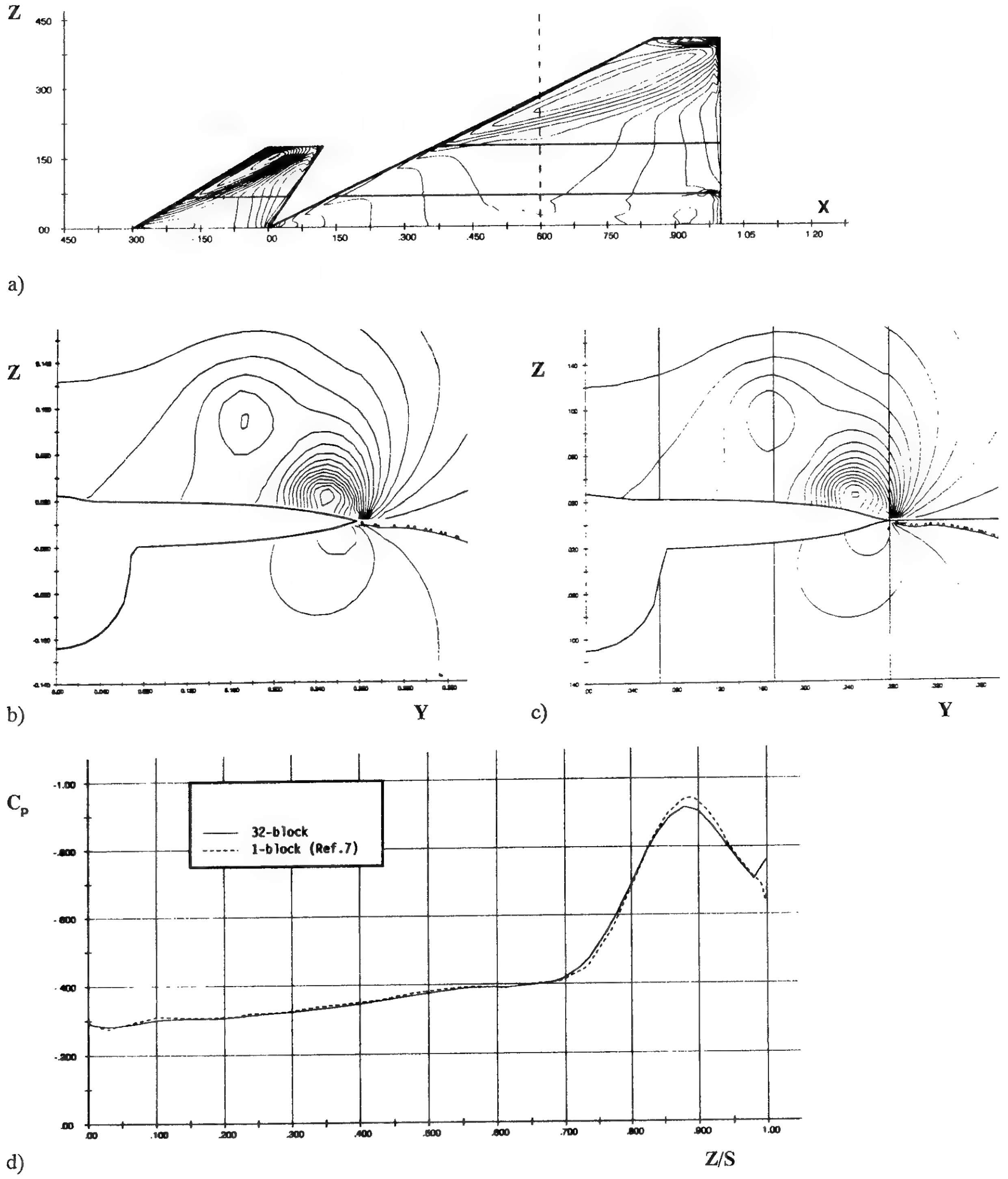


Figure 8 Flow around a wing-body-canard configuration (Mach = 0.85, α = 10°):a) iso-pressure plot on wing and canard; b) iso-pressure plot at 0.6 wing chords from a single-block solution [7]; c) iso-pressure plot at 0.6 wing chords from a solution on a 32-block decomposition of the single-block grid; d) pressure coefficient on wing surface at 0.6 wing chords.

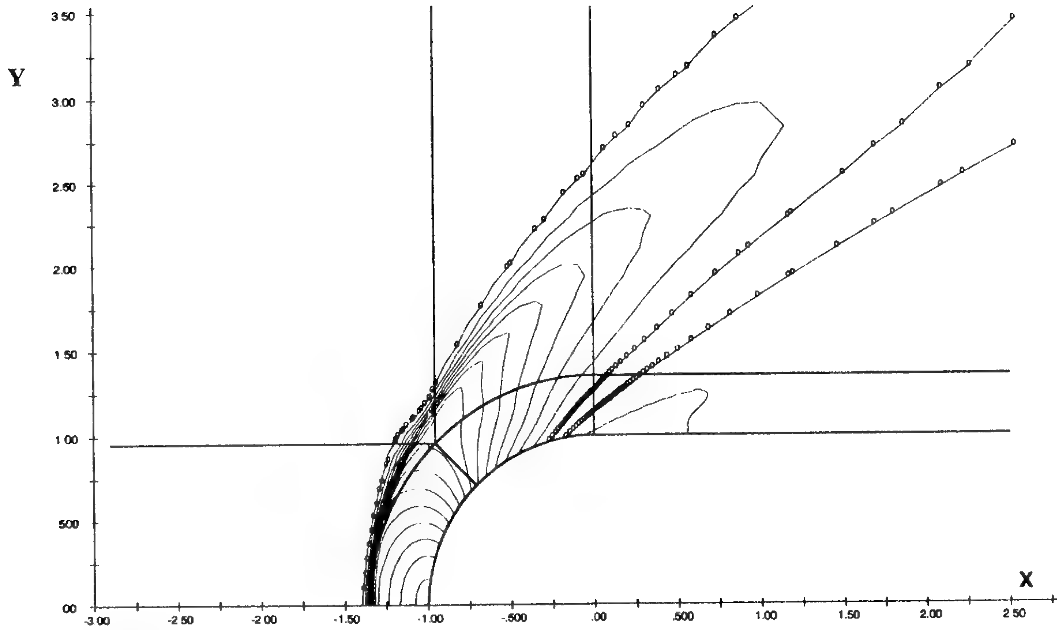


Figure 9 Iso-pressure plot of the flow around a blunt nosed body (Mach=2, $\alpha=0$): multiblock solution on a 7-block grid with locally unstructured topology.

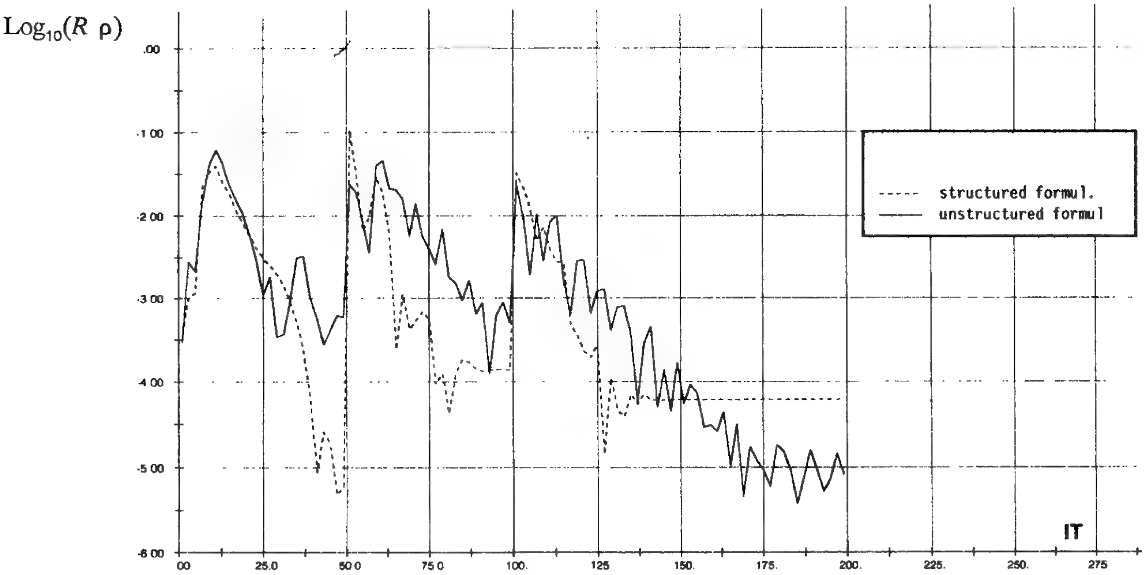


Figure 10 Convergence history of density residual on the locally unstructured edge of the 7-block grid around a blunt nosed body for solution of the flow at Mach=2 and $\alpha=0$:behaviour of different artificial dissipation models.

AMELIORATIONS RECENTES DU CODE DE CALCUL D'ECOULEMENTS COMPRESSIBLES FLU3M

L.Cambier,D.Darracq[1],M.Gazaix,Ph.Guillen,Ch.Jouet, L.Le Toullec
ONERA, B.P 72, 92322 Châtillon Cedex, France.

1

Abstract

We present three developments which have been introduced in the code FLU3M.

A numerical method for solving the unsteady Euler equations with time-varying rigid grids is first studied; it uses the van Leer scheme together with a second order in time implicit algorithm.

A bidimensional nozzle and an afterbody shape have been calculated with the Jones-Launder k-ϵ model, the implementation of which in the code is described for one and two species gases.

Then a new implicit algorithm is shown; precisely the DDLU factorization enables a reduction both in CPU time and in cost memory against the ADI factorization.

Résumé

Trois développements effectués dans le code FLU3M sont présentés. Une méthode de résolution des équations d'Euler instationnaires pour des mouvements de solide est d'abord étudiée; elle utilise le schéma de van Leer, ainsi qu'une approche implicite d'ordre deux en temps permettant de réduire les temps de calcul.

Une tuyère bidimensionnelle ainsi qu'un arrière-corps ont été calculés avec le modèle de turbulence k-ϵ de Jones-Launder, dont on décrit l'implantation dans le code pour un écoulement monoespèce ou biespèce. Des comparaisons avec l'expérience sont effectuées.

Puis un nouvel algorithme de résolution implicite a été étudié; la factorisation DDLU permet des gains en temps de calcul et place mémoire par rapport à une factorisation ADI.

1. Introduction

Depuis 1987, un code de calculs aérodynamiques (FLU3M), multidomaines, multiespèces, est développé à la division de l'Aérodynamique Théorique 1, de l'ONERA.

En 1989, les principaux choix numériques et la structure informatique du code sont publiés au séminaire international de Boston [1]. Des calculs numériques Euler gaz parfait et gaz réel à l'équilibre y sont présentés sur des configurations multidomaines telles que la navette Hermès; les écoulements étant supersoniques, des techniques de marche en espace sont mises en œuvre. La possibilité de calculs en gaz biespèce est illustrée par un calcul de jet chaud.

Depuis, FLU3M a fourni la base de nombreux développements, autant dans le domaine des modélisations physiques, que dans celui des techniques numériques améliorant la précision et la rapidité des calculs.

Ainsi, les équations de Navier-Stokes, en régime laminaire, sont maintenant résolues numériquement. Le code a été éprouvé sur plusieurs cas de validation: par exemple, une rampe hypersonique 3D présentée au Workshop d'Antibes [2], ou encore une configuration ogive-cylindre avec écoulement tourbillonnaire [3]. Pour les écoulements hypersoniques, un nouveau diagramme de Mollier a été étudié [4]; en plus des propriétés thermodynamiques de l'air à l'équilibre, il fournit les viscosités et conductivités en vue de calculs Navier-Stokes.

De nouvelles possibilités de discrétisation en espace ont été explorées et en particulier les techniques de maillages chimères. Des calculs complexes (séparation de missile) peuvent être ainsi plus facilement réalisés [5]. Nous présentons ici plus en détail trois axes de développement. Ces développements, réalisés dans un code unique, sont facilités par la grande modularité du code et par la clarté de la structure arborescente.

Un axe de développement est lié à l'étude des phénomènes d'aéroélasticité. La mise en œuvre des équations d'Euler instationnaires en maillage mobile est présentée, ainsi que différentes approches d'ordre 2 en temps permettant de réduire les coûts de calcul.

Dans le cadre des activités sur les modèles de turbulence, nous décrivons l'introduction d'un modèle à deux équations de transport de type k-ϵ, pour un gaz parfait monoespèce ou un gaz biespèce.

Pour terminer, un nouvel algorithme de résolution du système implicite est présenté. Nous étudions la factorisation DDLU réduisant l'espace mémoire et le temps de calcul par rapport à une factorisation ADI.

2. Calculs instationnaires en maillage mobile

Dans le cadre des études d'aéroélasticité pour les lanceurs de type Ariane, une méthode numérique Euler instationnaire a été développée dans FLU3M. Après la formulation des équations instationnaires, les dif-

[1] Doctorant sous convention CIFRE SNECMA

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

férents choix numériques sont exposés, puis le calcul d'un profil NACA en mouvement de tangage est présenté, ainsi que celui d'un sphère-cône 3D en oscillation autour de son centre de gravité.

2.1 Equations instationnaires

Considérons un profil γ en mouvement de tangage, muni d'un maillage $\Omega(t)$. Dans le cadre de cette étude, ce domaine $\Omega(t)$, supposé non déformable, est en mouvement par rapport à un repère absolu $\mathcal{R}_0$.

Sur $\Omega(t)$, les équations d'Euler s'écrivent, sous forme de lois de conservation :

$$\frac{d}{dt}\int_{\Omega(t)} W(\overrightarrow{x},t).d\tau + \int_{\partial\Omega(t)} F(W,\overrightarrow{n}).dS = \overrightarrow{0} \qquad (1)$$

avec $W = (\rho, \rho\overrightarrow{v}, \rho E)$, variables conservatives (ρ : densité, $\overrightarrow{v}$: vitesse absolue, E : énergie totale)

et

$$\text{où } F(W;\overrightarrow{n}) = \begin{pmatrix} \rho\overrightarrow{v_r}.\overrightarrow{n} \\ \rho\overrightarrow{v}.(\overrightarrow{v_r}.\overrightarrow{n}) + p\overrightarrow{n} \\ \rho E(\overrightarrow{v_r}.\overrightarrow{n}) + p\overrightarrow{v}.\overrightarrow{n} \end{pmatrix} \qquad (2)$$

$\overrightarrow{v_e}$ est la vitesse d'entraînement, $\overrightarrow{v_r}$ la vitesse relative. A la différence d'autres approches utilisant les vitesses relatives, les variables de calcul sont les variables absolues, c'est-à-dire les vitesses absolues exprimées dans le repère absolu $\mathcal{R}_0$. C'est une approche classique qui, par rapport aux équations en maillage fixe, demande une modification des flux numériques qui font intervenir la vitesse d'entraînement $\overrightarrow{v}_e$, ainsi qu'un calcul de métrique variable au cours du temps. Pour discrétiser les flux, nous utilisons les méthodes de décentrement; Vinokur en donne une analyse détaillée dans [7].

2.2 Discrétisation des flux

On pourra vérifier que le jacobien des flux a pour expression :

	ρ	$\rho\overrightarrow{v}$	ρE
f_ρ	$-\overrightarrow{v_e}.\overrightarrow{n}$	$\underline{n}$	0
$f_{\rho\overrightarrow{v}}$	$(\gamma-1)v^2.\overrightarrow{n} - v_n.\overrightarrow{v}$	$-(\gamma-1)\overrightarrow{n}\otimes\underline{v}$ $+\overrightarrow{v}\otimes\underline{n} + (v_n - v_{en})I$	$(\gamma-1)\overrightarrow{n}$
$f_{\rho E}$	$(\gamma-1)v^2 v_n - Hv_n$	$H\underline{n} - (\gamma-1)v_n\underline{v}$	$\gamma v_n - v_{en}$

avec $v_n = \overrightarrow{v}.\overrightarrow{n}$ et $v_{en} = \overrightarrow{v_e}.\overrightarrow{n}$. Les valeurs propres du jacobien des flux sont les suivantes :

$$\begin{cases} \lambda = \overrightarrow{v}_r.\overrightarrow{n} \ \ (ordre\ 3) \\ \lambda = \overrightarrow{v}_r.\overrightarrow{n} + c \ \ (ordre\ 1) \\ \lambda = \overrightarrow{v}_r.\overrightarrow{n} - c \ \ (ordre\ 1) \end{cases} \qquad (3)$$

Les vecteurs propres du jacobien des flux sont exactement les mêmes que dans le cas des équations en maillage fixe.

Pour discrétiser les flux, la décomposition de van Leer peut être étendue aux équations d'Euler instationnaires. Van Leer décompose le flux sous la forme suivante :

$f = f^+ + f^-$, où $f^+(resp.f^-)$ a des valeurs propres positives (resp. négatives).

En introduisant le nombre de Mach *relatif* normal $M_{rn} = \frac{\overrightarrow{v_r}.\overrightarrow{n}}{c}$, nous avons :

- Si $M_{rn} > 1$, $f^+ = f$
- Si $M_{rn} < -1$, $f^- = f$
- Pour $\mid M_{rn} \mid < 1$, f^+ a pour expression :

$$\begin{cases} f_e^+ \ = \ \frac{\rho c}{4}(M_{rn}+1)^2 \\ f^+_{\rho\overrightarrow{v}} \ = \ f_e^+[(\gamma-1)v_{rn}+2c]/\gamma.\overrightarrow{n} + f_e^+(\overrightarrow{v_t}+\overrightarrow{v_{en}}) \\ f^+_{\rho E} \ = \ f_e^+[(\gamma-1)v_{rn}+2c]^2/[2(\gamma^2-1)] \\ \qquad\quad +f_e^+[(\gamma-1)v_{rn}+2c]/\gamma.v_{en} \\ \qquad\quad +\frac{1}{2}f_e^+(\overrightarrow{v_t}+\overrightarrow{v_{en}})^2 \end{cases} \qquad (4)$$

Le flux de van Leer s'écrit alors :

$$f_{van\ Leer}(W_g, W_d) = f^+(W_g) + f^-(W_d)$$

Pour l'utilisation de méthodes implicites, ce flux doit être linéarisé.

2.3 Métrique instationnaire

Au cours du mouvement du profil, le maillage est mobile et, par conséquent, les normales aux interfaces doivent être calculées à chaque instant. Par hypothèse, le maillage ne se déforme pas, les volumes ne changent donc pas. Ils sont calculés une fois pour toutes à l'instant initial t_0.

Il faut évaluer la moyenne de $\overrightarrow{n}$, sur un pas de temps Δt, ce que l'on fait en considérant l'instant $t_{n+\frac{1}{2}}$. Nous avons alors : $\overrightarrow{n}(t_{n+\frac{1}{2}}) = R(t_{n+\frac{1}{2}}).\overrightarrow{n}(t_0)$, où R est la matrice de rotation du mouvement prise à l'instant $n+\frac{1}{2}$. Le calcul des flux nécessite également la connaissance, à l'instant $t_{n+\frac{1}{2}}$, de la vitesse d'entraînement aux interfaces des mailles de calcul. Connaissant les coordonnées à l'instant t=0 du centre I d'une interface, la vitesse d'entraînement est donnée par :

$$\overrightarrow{v_{eI}}(t_{n+\frac{1}{2}}) = \overrightarrow{v_A}(t_{n+\frac{1}{2}}) + \bar{\omega}(t_{n+\frac{1}{2}}).\overrightarrow{AI}$$

où $\bar{\omega}$ est le tenseur de vitesse de rotation du solide, A un point du solide.
Ceci complète et définit les données nécessaires à un calcul instationnaire, en maillage rigide.

2.4 Conditions aux limites instationnaires

Considérons l'exemple d'un profil muni d'un maillage en C, en mouvement par rapport à $\mathcal{R}_0$. Un écoulement uniforme est imposé à l'amont du profil.

Deux conditions aux limites sont à envisager: d'une part, aux frontières à l'infini, pour imposer un écoulement tout en évitant les réflexions d'ondes; d'autre part, à la paroi où une condition de glissement doit être imposée.

1. *Conditions à la limite à la paroi.*

D'après la condition de glissement, $\vec{v_r}.\vec{n} = 0$, où $\vec{v_r}$ est la vitesse relative. Le flux à la paroi devient :

$$F(W;\vec{n}) = S_l.\begin{pmatrix} 0 \\ p.\vec{n} \\ p\vec{v_e}.\vec{n} \end{pmatrix}$$

où p est la pression à l'interface, calculée par une extrapolation, éventuellement corrigée par une relation caractéristique.

2. *Conditions à la limite d'entrée-sortie.*

Nous adoptons la formulation proposée par Collercandy [9]. Elle est étendue aux équations en maillage mobile.
Cinq caractéristiques de pentes λ, $\lambda \in \{v_{rn}, v_{rn}+c, v_{rn}-c\}$ arrivent à l'interface à l'instant n+1. Suivant le signe de la pente λ, la variable caractéristique associée sera calculée avec un état extérieur ou intérieur.
Plus précisément, le calcul des valeurs propres est effectué à l'aide d'un état moyen:
$W_m = \frac{1}{2}(W_{interieur} + W_{exterieur})$, ce qui permet de calculer les valeurs propres et de connaître leur signe. On calcule ensuite les variables caractéristiques associées à ces valeurs propres, avec les états intérieur et extérieur .
Si λ est négative, c'est la variable caractéristique extérieure qui sera choisie; sinon, on prendra la variable caractéristique intérieure.

2.5 Discrétisation en temps

La précision en temps des schémas utilisés en instationaire est un point important. Plusieurs schémas d'ordre un ou deux en temps sont décrits et leurs propriétés de stabilité sont brièvement rappelées. L'équation à discrétiser en temps est la suivante:

$$\frac{\partial W}{\partial t} = g(W) \qquad (5)$$

2.5.1 Schémas précis à l'ordre 1

Le schéma explicite : $W^{n+1} = W^n + \Delta t g^n$ est précis à l'ordre 1. Il est stable sous la condition $cfl \preceq 1$, ce qui, en pratique impose des pas de temps très petits. En effet, pour calculer avec un maillage assez fin un profil NACA64A010 oscillant avec un mouvement de battement à une fréquence de 100 Hz, 40 000 pas de temps explicites sont nécessaires pour effectuer un cycle.

Le schéma implicite : $W^{n+1} = W^n + \Delta t g^{n+1}$ est précis à l'ordre 1. La fonction g^{n+1} est linéarisée en : $g^{n+1} = g^n + \Delta t A^n(W^{n+1} - W^n)$ ce qui conduit au schéma : $(I - (\Delta t)^2 A^n)(W^{n+1} - W^n) = g^n \Delta t$. Ce schéma est inconditionnellement stable. Pour obtenir la même précision qu'un calcul explicite, sur des grandeurs telles que la portance et le moment, des *cfl* de l'ordre de 100 peuvent être utilisés pour les profils oscillants.

2.5.2 Schémas précis à l'ordre 2

Schéma Runge-Kutta implicite de Iannelli-Baker
C'est un schéma Runge-Kutta à deux étapes implicites [10].

$$(I - \alpha(\Delta t)^2 A^n)\Delta W1 = g(W^n)\Delta t \qquad (6)$$

$$(I - \alpha(\Delta t)^2 A^n)\Delta W2 = g(W^n \; + \; \beta\Delta W1)\Delta t \qquad (7)$$

$$et \;\; W^{n+1} \; = \; W^n \; + \; \gamma_1 \Delta W1 \; + \; \gamma_2 \Delta W2 \qquad (8)$$

$$avec \;\alpha \; = \; \frac{2-\sqrt{2}}{2}, \; \beta \; = \; 2(3\sqrt{2}-4), \qquad (9)$$

$$\gamma_1 \; = \; \frac{6-\sqrt{2}}{8}, \gamma_2 \; = \; \frac{6+\sqrt{2}}{8} \qquad (10)$$

Ce schéma est inconditionnellement stable, avec des *cfl* beaucoup plus grands que ceux utilisés avec un schéma implicite d'ordre 1, de l'ordre de 400, toujours pour obtenir des résultats de précision équivalente à celle d'un calcul explicite.

2.6 Cas de validation

2.6.1 Profil NACA64A010

Le profil choisi est un NACA64A010, correspondant aux conditions expérimentales suivantes (Fig.1):

$\mathcal{M}$	0.796
p^t_∞	203321 Pa
α_0	1.01°
f	34.4 Hz
$k = \frac{\omega L_{ref}}{V_{inf}}$	0.404
$\frac{x_\alpha}{L_{ref}}$	0.25

La loi du mouvement du profil de l'aile est donnée par:

$$\alpha(t) \;\; = \;\; \alpha_0 * sin(k.\tau)$$

Le profil a été défini dans un rapport AGARD [11] et calculé par de nombreux auteurs [8].

Pour chaque essai instationnaire, on présente l'évolution de la portance et le moment. Les calculs des différentes approches en temps sont comparés à l'expérience et à un calcul explicite de référence à *cfl* =0,8.

Sur les figures 2 et 3, les calculs des approches d'ordre 1 et d'ordre 2 en temps, pour un *cfl* de 400, sont représentés. Si nous considérons le calcul du moment, l'approche d'ordre 1 ne donne pas le même résultat que le calcul de référence explicite, alors que l'approche d'ordre 2 donne un résultat identique.

A *cfl* = 100, pour la méthode d'ordre 1, le coût de calcul est divisé par 17 par rapport à une méthode explicite à *cfl* = 1 (400 pas de temps/cycle contre 40 000 en explicite). Le coût de la méthode Runge-Kutta implicite d'ordre 2 est pratiquement identique à celui de la méthode d'ordre 1, puisque la matrice implicite est la même dans les deux pas de calcul.

En conclusion, la méthode implicite d'ordre 2 en temps est la plus indiquée pour des calculs instationnaires.

2.6.2 Sphère-cône Aérospatiale

Un corps sphéro-conique 3D a également été calculé (Fig.4); ce corps est en oscillation autour de son centre de gravité G. Le nombre de Mach à l'infini est de 7. L'angle de tangage maximum est $\alpha_0 = 1^o$ La loi du mouvement est donnée par:

$$\alpha(t) \quad = \quad \alpha_0 * sin(k.\tau)$$

avec $k = \frac{\Omega L}{V_{inf}} = 0,386$. Un calcul instationnaire à *cfl* = 100, avec la méthode Runge-Kutta implicite a été réalisé (Fig.4). Une comparaison a été faite avec un autre calcul effectué par Aérospatiale [?]. Un écart de 8% est observé sur $C_{m\alpha}$, alors que $C_{m\alpha\cdot} + C_{mq}$ est identique dans les deux calculs (Fig.5).

3. Modélisation de la turbulence

3.1 Modèle à deux équations de transport

Le modèle de turbulence (k, ϵ) de Jones-Launder [13] est implanté dans le code de calcul FLU3M. Les deux équations de transport pour ρk et $\rho\epsilon$ s'écrivent:

$$\partial_t \rho k + div(\rho k \vec{v}) = \bar{\bar{\tau}}_R : \nabla \vec{v} + div((\mu + \frac{\mu_t}{\alpha_k})\nabla k) - \rho\epsilon + D_k \tag{11}$$

$$\partial_t \rho\epsilon + div(\rho\epsilon\vec{v}) = C_{\epsilon_1}\frac{\epsilon}{k}\bar{\bar{\tau}}_R : \nabla \vec{v} + div((\mu + \frac{\mu_t}{\alpha_\epsilon})\nabla\epsilon) - C_{\epsilon_2} f_2 \frac{\epsilon^2}{k}\rho\epsilon + D_\epsilon \tag{12}$$

Le coefficient de viscosité turbulente μ_t a pour expression:

$$\mu_t = C_\mu f_\mu \frac{\rho k^2}{\epsilon} \tag{13}$$

Le choix suivant a été considéré pour les coefficients du modèle: $C_{\epsilon_1} = 1.57, C_{\epsilon_2} = 2., \alpha_k = 1., \alpha_\epsilon = 1.3, C_\mu = 0.09$. Les termes D_k et D_ϵ désignent des termes additionnels liés à la formulation bas-Reynolds et destinés à représenter l'amortissement de la turbulence au voisinage des parois. Dans l'article de Jones et Launder [13], l'expression de ces termes est donnée en repère de couche limite. Dans le cadre de la résolution des équations de Navier-Stokes, nous avons considéré les expressions suivantes pour D_k et D_ϵ:

$$D_k = -2\mu||\nabla\sqrt{k}||^2 \tag{14}$$

$$D_\epsilon = \frac{2\mu\mu_t}{\rho}||rot(rot\vec{v})||^2 \tag{15}$$

Les quantités f_2 et f_μ sont aussi liées à l'amortissement de la turbulence près des parois. Elles sont fonctions du nombre de Reynolds turbulent Re_t:

$$Re_t = \frac{\rho k^2}{\mu\epsilon} \tag{16}$$

$$f_2 = 1 - 0.3exp(-Re_t^2) \tag{17}$$

$$f_\mu = exp(-\frac{2.5}{1 + \frac{Re_t}{50}}) \tag{18}$$

Ce choix de fonctions d'amortissement ne faisant intervenir ni la distance à la paroi, ni le frottement pariétal, permet de réaliser une programmation du modèle de turbulence indépendante de l'application considérée, ce qui constitue un avantage important pour un code traitant des applications multidomaines complexes, tel que le code FLU3M.

Le modèle de turbulence (k, ϵ) qui vient d'être décrit peut être associé dans le code FLU3M, soit à une formulation monoespèce, soit à une formulation biespèce. Dans cette dernière formulation, on se place alors dans le cadre de l'écoulement compressible turbulent d'un mélange non réactif de deux espèces, chaque espèce étant supposée être un gaz parfait à chaleurs spécifiques constantes. La discrétisation des équations s'effectue de manière analogue pour le système "monoespèce (k, ϵ)" et pour le système "biespèce (k , ϵ)". Les flux convectifs sont discrétisés à l'aide d'extensions du solveur de Riemann de Roe aux systèmes d' équations couplées. Les valeurs propres de la matrice Jacobienne ont pour expression: $\lambda_1 = u$ (ordre: $neq - 2$), $\lambda_2 = u + c$ (ordre: 1), $\lambda_3 = u - c$ (ordre: 1), où *neq* désigne le nombre total d'équations (7 en monoespèce et 8 en biespèce). La quantité c est une vitesse du son modifiée donnée par :

$$c^2 = \gamma(\frac{p}{\rho} + \frac{2}{3}k) \tag{19}$$

Le rapport des chaleurs spécifiques γ est supposé constant dans la formulation monoespèce, alors qu'en biespèce, il dépend des densités partielles et des chaleurs spécifiques des deux espèces, de la manière suivante:

$$\gamma - 1 = \frac{\rho_1 Cv_1(\gamma_1 - 1) + \rho_2 Cv_2(\gamma_2 - 1)}{\rho_1 Cv_1 + \rho_2 Cv_2}, (20)$$

Le flux numérique de Roe s'écrit:

$$F^R(U_L, U_R) = \frac{1}{2}(F(U_L) + F(U_R)) - \frac{1}{2} P^R \|\Lambda^R\| P^{R^{-1}} (U_R - U_L) \quad (21)$$

où Λ^R désigne la matrice diagonale des valeurs propres, et P^R et $P^{R^{-1}}$ désignent les matrices de passage. La notation R en indice supérieur indique que les quantités sont calculées à l'aide de moyennes de Roe.

La précision du second ordre en espace est obtenue grâce à l'approche MUSCL appliquée sur les variables primitives. Les flux visqueux sont évalués à l'aide d'une discrétisation centrée en espace. Dans le cas de la formulation biespèce, on tient compte de la diffusion entre les espèces par l'intermédiaire d'un nombre de Lewis Le et d'un nombre de Lewis turbulent Le_t . Une accélération de convergence est réalisée à l'aide d'une phase implicite et de la technique du pas de temps local. La phase implicite s'appuie sur une linéarisation des flux de van Leer pour les flux convectifs, une linéarisation similaire à celle de Coakley pour les flux visqueux, une linéarisation simplifiée de la partie négative des termes source et une inversion ADI de la matrice implicite.

A titre d'exemples, on présente ici les résultats obtenus dans le cadre de la formulation "monoespèce (k, ϵ)" , sur une configuration d'interaction onde de choc/couche limite dans un canal bidimensionnel, puis sur une configuration d'arrière-corps axisymétrique. La première configuration correspond à une expérience [14] réalisée à l'ONERA dans une tuyère symétrique. Le nombre de Mach en amont de l'interaction est égal à 1.45. Sur la figure 6 qui représente les courbes iso-nombre de Mach calculées, on peut voir la structure classique de choc en λ dans la région d'interaction et l'important épaississement de la couche limite résultant de l'interaction avec le choc. La figure 7 présente une comparaison avec l'expérience portant sur la distribution de pression pariétale. Le plateau de pression obtenu par le calcul dans la région d'interaction est plus petit que dans l'expérience, ce qui correspond à une légère sous-estimation de la taille de la région décollée. Le résultat obtenu est sur ce point comparable à des résultats obtenus antérieurement avec d'autres codes de calcul mettant en oeuvre le modèle (k, ϵ), sur la même configuration.

La deuxième configuration traitée correspond à un arrière-corps axisymétrique muni d'une tuyère. Les conditions de l'écoulement externe sont les suivantes: nombre de Mach égal à 4.18, température et pression génératrices respectivement égales à 325 Kelvins et 10 bars. La pression génératrice du jet est plus élevée et égale à 42.3 bars. Le nombre de Reynolds calculé à partir des grandeurs critiques associées au jet et du rayon du culot est égal à 1.15 10^7. Le domaine de calcul est divisé en trois sous-domaines: un sous-domaine D_1 correspondant à l'écoulement externe, un sous-domaine D_3 correspondant à la sortie de la tuyère et au jet, et un sous-domaine intermédiaire D_2 comprenant la région du culot. Le nombre total de points de maillage est égal à 13,879. Des raffinements importants sont introduits près des parois. Par exemple, la taille de maille près de la paroi externe de l'arrière-corps est égale à $10^{-4}R$. Sur la frontière amont du sous-domaine externe D_1, on impose des profils issus des données expérimentales pour la vitesse et les grandeurs turbulentes, alors que, sur la frontière amont du sous-domaine D_3, les profils imposés sont issus d'un calcul préliminaire de l'écoulement dans la tuyère.

La figure 8 qui représente la solution sous forme de courbes iso-nombre de Mach, montre la forme classique en tonneau du jet, ainsi que l'onde de choc située dans le jet. Une comparaison avec l'expérience [15] est représentée sur les figures 9 et 10, sous forme de profils d'énergie cinétique de turbulence et de pression pitot dans deux sections situées en aval du culot à des distances égales à $0.59R$ et à 6 R. Les points expérimentaux ont été obtenus par vélocimétrie laser et par un tube de Pitot. Bien que les données expérimentales pour l'énergie cinétique de turbulence ne soient relatives qu'à la partie externe de l'écoulement, l'accord apparaît comme satisfaisant.

4. Factorisation DDLU

4.1 Description de l'algorithme

L'analyse de stabilité linéaire de von Neumann de la factorisation approchée ADI révèle une instabilité inconditionnelle en 3D (Cf. Ying [21]). Même si les termes non linéaires jouent un rôle stabilisateur comme tendent à le prouver les codes de calcul utilisant une telle approche, la factorisation triple ADI reste pénalisée par un nombre d'opérations important et surtout une sévère restriction de cfl due à l'erreur de factorisation en Δt^3. Une factorisation de type DDLU a donc été développée pour améliorer l'efficacité de l'algorithme implicite. Les premières méthodes de décomposition DDLU de la matrice implicite ont été proposées simultanément par Jameson et Turkel [17] et Steger et Warming [20] en 1981. Alors que les techniques de directions alternées consistent à substituer à l'opérateur implicite un opérateur factorisé suivant les directions du maillage, la méthode LU le décompose en deux matrices triangulaires supérieure et inférieure. Jameson et Turkel montrent qu'un tel système est bien conditionné si les matrices sont à diagonales dominantes. Aussi ont-ils proposé une décomposition des matrices jacobiennes de la matrice implicite augmentant la diagonale. Dans le schéma original, le système implicite est mis sous la forme :

$$LU.\Delta\widehat{Q} = -\Delta t\ \widehat{R} \qquad (22)$$

Dans le cas d'une discrétisation décentrée, il vient :

$$\begin{cases} L = I + \Delta t(\partial_\xi^- \widehat{A}^+ + \partial_\eta^- \widehat{B}^+ + \partial_\zeta^- \widehat{C}^+) \\ U = I + \Delta t(\partial_\xi^+ \widehat{A}^- + \partial_\eta^+ \widehat{B}^- + \partial_\zeta^+ \widehat{C}^-) \end{cases} \qquad (23)$$

où $\widehat{A}^+, \widehat{A}^-, \widehat{B}^+, \widehat{B}^-, \widehat{C}^+, \widehat{C}^-$ sont les matrices jacobiennes des flux et ξ, η, ζ les coordonnées curvilignes. Ce schéma reste peu utilisé sous cette forme.
Jameson et Turkel [17] ayant montré que la condition de dominance diagonale permet d'assurer le bon conditionnement des facteurs L et U, Jameson et Yoon [18] ont développé une variante (que l'on appellera ici DDLU par analogie avec le DDADI), renforçant la diagonale pour les équations d'Euler, puis Yoon et Jameson [22] pour les équations de Navier-Stokes, Shuen et Yoon [19] pour les écoulements réactifs, enfin Darracq (1995) [16] pour le modèle k-ϵ . Avec ce formalisme, il vient:

$$LD^{-1}U.\Delta\widehat{Q} = -\Delta t\widehat{R} \qquad (24)$$

avec :

$$\begin{cases} L = I + \Delta t(\partial_\xi^- \widehat{A}^+ + \partial_\eta^- \widehat{B}^+ + \partial_\zeta^- \widehat{C}^+ - \widehat{A}^- - \widehat{B}^- - \widehat{C}^-) \\ D = I + \Delta t(\widehat{A}^+ - \widehat{A}^- + \widehat{B}^+ - \widehat{B}^- + \widehat{C}^+ - \widehat{C}^-) \\ U = I + \Delta t(\partial_\xi^- \widehat{A}^+ + \partial_\eta^- \widehat{B}^+ + \partial_\zeta^- \widehat{C}^+ - \widehat{A}^+ - \widehat{B}^+ - \widehat{C}^+) \end{cases} \qquad (25)$$

Les matrices jacobiennes $\widehat{A}^+$, $\widehat{B}^+$ et $\widehat{C}^+$ (respectivement $\widehat{A}^-$, $\widehat{B}^-$ et $\widehat{C}^-$) sont construites de façon à ce qu'elles ne possèdent que des valeurs propres positives (respectivement négatives), c'est-à-dire :

$$\begin{cases} \widehat{A}^+ = T_\xi \widehat{\Lambda}_\xi^+ T_\xi^{-1} & ; \quad \widehat{A}^- = T_\xi \widehat{\Lambda}_\xi^- T_\xi^{-1} \\ \widehat{B}^+ = T_\eta \widehat{\Lambda}_\eta^+ T_\eta^{-1} & ; \quad \widehat{B}^- = T_\eta \widehat{\Lambda}_\eta^- T_\eta^{-1} \\ \widehat{C}^+ = T_\zeta \widehat{\Lambda}_\zeta^+ T_\zeta^{-1} & ; \quad \widehat{C}^- = T_\zeta \widehat{\Lambda}_\zeta^- T_\zeta^{-1} \end{cases} \qquad (26)$$

On peut écrire de façon générale la décomposition des matrices jacobiennes sous la forme :

$$\widehat{\Lambda}^{\pm}_{\xi,\eta,\zeta} = \frac{1}{2}[\widehat{\Lambda}_{\xi,\eta,\zeta} \pm \gamma(\widehat{\Lambda}_{\xi,\eta,\zeta})] \qquad (27)$$

avec $\widehat{\Lambda}_{\xi,\eta,\zeta}$ matrice diagonale des valeurs propres $\widehat{\lambda}_{\xi,\eta,\zeta}$.
La fonction γ décrit le caractère du décentrement.

Dans la décomposition classique, on définit γ par:

$$\begin{aligned} \gamma(\widehat{\Lambda}_{\xi,\eta,\zeta}) &= |\widehat{\Lambda}_{\xi,\eta,\zeta}| \\ &= \mathrm{diag}(|\widehat{\lambda}_{\xi,\eta,\zeta}|) \end{aligned} \qquad (28)$$

Dans la décomposition de Jameson et Turkel [17] [2],on vise à augmenter la dominance diagonale :

$$\gamma(\widehat{\Lambda}_{\xi,\eta,\zeta}) = \beta.\max(|\widehat{\lambda}_{\xi,\eta,\zeta}|)I \qquad (29)$$

avec $\beta \geq 1$. Les relations (26) et (26) permettent d'écrire avec (27) :

$$D = I + \Delta t[T_\xi\gamma(\widehat{\Lambda}_\xi)T_\xi^{-1} + T_\eta\gamma(\widehat{\Lambda}_\eta)T_\eta^{-1} + T_\zeta\gamma(\widehat{\Lambda}_\eta)T_\zeta^{-1}] \quad (30)$$

La diagonale D possède une structure bloc dans le cas de la décomposition (28) mais devient scalaire quand on utilise (29) :

$$D = I + \Delta t\beta[\max(|\widehat{\lambda}_\xi|) + \max(|\widehat{\lambda}_\eta|) + \max(|\widehat{\lambda}_\zeta|)]I \; (31)$$

Remarquons que cette propriété de réduction de la diagonale bloc à une diagonale scalaire est vérifiée pour les factorisations de type DDLU, DDADI et même ADI. A l'opposé, la factorisation LU de base (23), à cause de sa nature dissymétrique, ne peut bénéficier de cette diagonalisation. Les matrices jacobiennes aux interfaces sont évaluées à partir de la moyenne de Roe afin de préserver une consistance avec le schéma explicite.

Balayage plan oblique

Le balayage du domaine de calcul suivant les directions diagonales (i+j+k constant) dans le sens croissant (opérateur L) puis décroissant (opérateur U) permet de vectoriser complètement l'inversion des matrices L et U en évitant les récurrences non vectorisables. La récurrence entre points de la factorisation ADI devient une récurrence entre plans lors de la factorisation DDLU.
En outre, le balayage diagonal fait intervenir, autour du point courant, des points dont la mise à jour a été effectuée à l'étape précédente. On peut ainsi les ajouter au membre de droite : il n'y a donc aucun bloc à inverser.

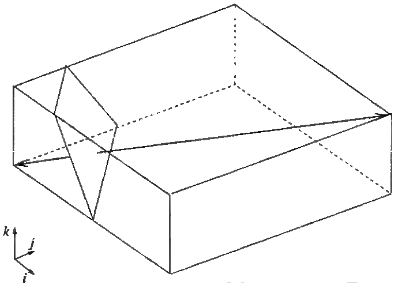


Fig.12 : Balayage oblique en 3D

A titre de comparaison, on a implanté un algorithme du type SSOR avec les choix de décomposition (29) conduisant à la diagonale scalaire (33). Il s'agit d'une

[2] Cette décomposition doit être distinguée de la technique du rayon spectral appliquée aux schémas décentrés [20] :

$$\widehat{A}^{\,\pm} = \max(|\widehat{\lambda}_\xi^{\pm}|)I$$

approche itérative de sur-relaxation symétrique avec balayage oblique :

$$\begin{aligned}(D+\omega L)\Delta\widehat{Q}^{k+\frac{1}{2}} &= -\omega\Delta t\widehat{R}+(1-\omega)D\widehat{Q}^{k}-\omega U\widehat{Q}^{k}\\(D+\omega U)\Delta\widehat{Q}^{k+1} &= -\omega\Delta t\widehat{R}+(1-\omega)D\widehat{Q}^{k+\frac{1}{2}}-\omega L\widehat{Q}^{k+\frac{1}{2}}\end{aligned}\qquad(32)$$

4.2 Application

On présente ici l'application de l'algorithme DDLU dans sa version fluide parfait sur le cas test d'un fuselage lenticulaire avec rétreint. Le nombre de Mach vaut 4,5 et l'incidence est de 10°. Le maillage est composé de 42 × 27 × 44 points. La figure 11 représente la solution obtenue à partir du schéma DDLU. La solution donnée par le schéma ADI est identique. La figure 13 représente l'histoire de la convergence des résidus implicites de la densité. La montée en CFL est effectuée jusqu'à une valeur de 500 pour les formulations DDLU et SSOR et jusqu'à 100 pour la formulation ADI. La vitesse de convergence de l'implicite DDLU est meilleure que celle de l'implicite ADI. L'algorithme SSOR permet une convergence plus rapide que celle de l'algorithme DDLU. Mais le nombre d'itérations internes, une douzaine, augmente les temps de calcul qui deviennent comparables à ceux du schéma ADI. Le tableau 1 donne le temps de calcul par point et par itération et le nombre de tableaux 3D nécessaires pour le stockage de la matrice implicite, pour les décompositions ADI et DDLU.

La version DDLU est 2,3 fois plus rapide que la version ADI, et, d'autre part, le schéma LU requiert près de 1,5 fois moins de place mémoire pour le stockage des matrices implicites. En effet, on ne stocke en chaque point courant qu'un seul vecteur D_{ijk} alors que la factorisation ADI demande la réservation mémoire en chaque point de trois blocs $D_{\xi_{ijk}}$, $D_{\eta_{ijk}}$ et $D_{\zeta_{ijk}}$. De plus, la factorisation ADI fait appel à des tableaux temporaires lors de l'inversion.

Algorithme	Temps CPU (μs)	Mémoire
ADI Euler 3D	49	225
DDLU Euler 3D	21	155

Tableau 1: Comparaison des temps de calcul

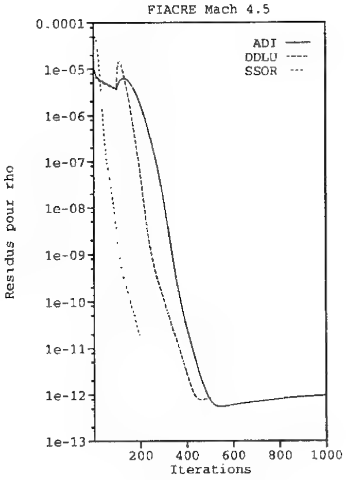


Fig.13: Comparaison des vitesses de convergence des algorithmes

5. Conclusion

Trois développements récents dans le code FLU3M ont été présentés.

Les équations d'Euler instationnaires (en maillage non déformable mobile) ont été discrétisées avec un schéma Runge-Kutta implicite d'ordre 2 en temps associé aux flux de van Leer. Les cas de validation présentés, 2D et 3D, montrent la précision et la rapidité de la méthode, aussi précise qu'un calcul explicite, mais 70 fois plus rapide.

L'implémentation du modèle k-ϵ de Jones- Launder, y compris pour un gaz biespèce, a ensuite été décrite. Nous utilisons le solveur de Roe pour résoudre le système complet des équations de Navier-Stokes couplées avec les équations de transport pour k et ϵ. Si des résultats satisfaisants ont été obtenus, notamment sur un cas d'arrière-corps, des études concernant l'application du modèle restent à effectuer; en particulier, le champ initial des variables k et ϵ doit être déterminé pour commencer le calcul; de plus, des phénomènes de relaminarisation peuvent apparaître. Pour terminer, l'algorithme implicite de décomposition DDLU permet de réduire les coûts mémoire et temps de calcul à chaque itération. Cet algorithme peut être étendu aux équations avec modèle k-ϵ.

Remerciements: Les développements relatifs au modèle k-ϵ, ainsi que ceux relatifs à l'instationnaire, ont été soutenus par Aérospatiale Espace et Défense et par le CNES. Les travaux sur l'algorithme DDLU ont été effectué dans le cadre de la thèse de D. Darracq, stagiaire CIFRE ONERA-SNECMA.

References

[1] Ph. Guillen, M. Dormieux, *Design of a 3D Multi-Domain Euler Code*, International Seminar on Supercomputing, Boston, Oct. 3-5, 1989.

[2] C. Jouet, M. Borrel, *Navier-Stokes Computation over a Three- Dimensional Ramp*, Proceedings of Hypersonic Workshop of Antibes, avril 1991.

[3] C. Jouet, P. D'Espiney ,*3D Laminar and 2D Turbulent Computations with the Navier-Stokes Solver FLU3M*, . 8th Int. Conf. on Numerical Methods in Laminar and Turbulent Flow, Swansea, July 1993.

[4] M. Gazaix, *Hypersonic Inviscid and Viscous Flow Computations with a New Optimized Thermodynamic Equilibrium Model*, AIAA 93-0893, Reno, Jan. 1993.

[5] J.P. Gillyboeuf, P. Mansuy, S. Pavsic, *Two New Chimera Methods : Application to Missile Separation*, AIAA paper, Reno, Jan. 1995.

[6] C. Jouet, V. Cheret, *Méthode numérique appliquée à des profils oscillants* , 31ème colloque AAAF d'aérodynamique appliquée, mars 1995.

[7] M. Vinokur, *An Analysis of Finite-Difference and Finite-Volume Formulations of Conservation Laws*, Journal of Computational Physics, Vol. 81, *n°* 1, mars 1989.

[8] J. Sidès, *Computation of Unsteady Transonic Flows with an Implicit Numerical Method for Solving the Euler Equations*, La Recherche Aérospatiale, *n°*2, février 1985.

[9] R. Collercandy, *An Improved Approach for the Computation of Transonic/Supersonic Flows with Applications to Aerospace Configurations*, AIAA paper 92-2613 CP, Jan. 1992.

[10] G.S. Iannelli, A.J. Baker, *A Stiffly Stable Implicit Runge-Kutta Algorithm For CFD Applications*, AIAA paper 88-416, Jan. 1988.

[11] R.J. Zwaan, *Naca 64A006 oscillating flap* , AGARD Report n. 702, Compendium of unsteady aerodynamic measurements, August 1982.

[12] F.
Coron, F. Ruffino , *Prévision de l'aérodynamique instationnaire autour de lanceurs pour l'étude de l'aéroélasticité* , 31ème Colloque AAAF d' Aérodynamique Appliquée, Mars 1995.

[13] W.P. Jones et B.E. Launder, *The Prediction of Laminarization with a Two-Equation Model of Turbulence*, Int.J.of Heat and Mass Transfer, Vol.15, pp.301-314 (1972).

[14] J. Délery, *Experimental Investigation of Turbulence Properties in Transonic Shock/Boundary Layer Interaction*, AIAA Journal, vol.21, pp.180-185, Fév. 1983.

[15] P. Reijasse, *Modélisation de l'écoulement supersonique autour de l'arrière-corps du lanceur ARIANE 5: Expériences de validation de code sur des configurations d'arrière-corps axisymétriques*, ONERA RF No 1/2536AY (1991).

[16] Darracq D., *Etude numérique des interactions choc-choc et choc-couche limite turbulente en régime hypersonique* , Thèse de Doctorat de l'Université de Poitiers, France (1995).

[17] Jameson A., Turkel E., *Implicit Schemes and LU Decomposition* , *Mathematics of Computation*, Vol. 37, n°156, pp 385-397, (1981).

[18] Jameson A., Yoon S., *LU Implicit Schemes with Multiple Grids for the Euler Equations*, *AIAA Paper* 86-0105, (1986).

[19] Shuen J.S., Yoon S., *A Numerical Study of Chemically Reacting Flows Using an LU-SSOR Scheme* , *AIAA Paper* 88-0436, (1988).

[20] Steger J.L., Warming R.F., *Flux Vector Splitting of the Inviscid Gas Dynamics Equations with Application to Finite-Difference Methods*, *Jour. Comp. Phys.*, Vol. 40, pp 263-293, (1981).

[21] Ying S., *Three-Dimensional Implicit Approximately Factorized Schemes for Equations in Gas Dynamics*, Ph.D. Thesis, Stanford University, Stanford, CA, (1986).

[22] Yoon S., Jameson A., *A LU-SSOR Scheme for the Euler and Navier-Stokes Equations*, *AAIA Paper* 87-600, (1987).

Calculs k-ε

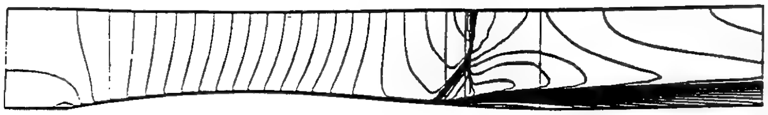


FIG. 6 - *Tuyère de Delery : lignes iso-nombre de Mach*

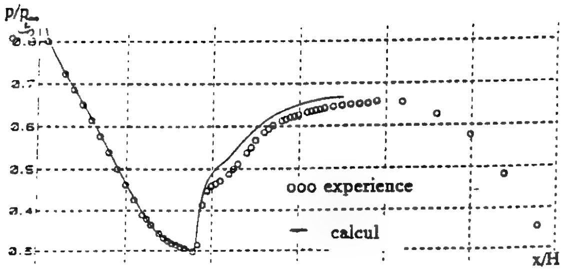


FIG. 7 - *Tuyère de Delery : profil de pression à la paroi*

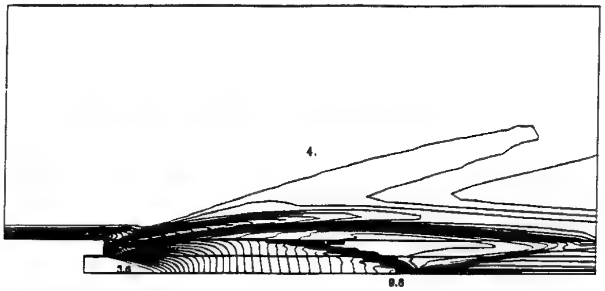


FIG. 8 - *Arrière-corps: lignes iso-nombre de Mach*
$\Delta M = 0,2$

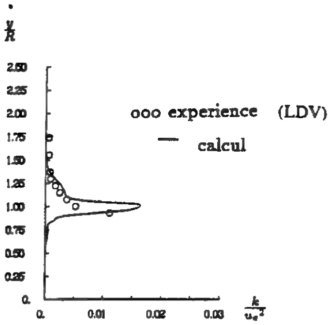


FIG. 9 - *Arrière-corps: profil d'energie cinétique de turbulence à x/R=0,59*

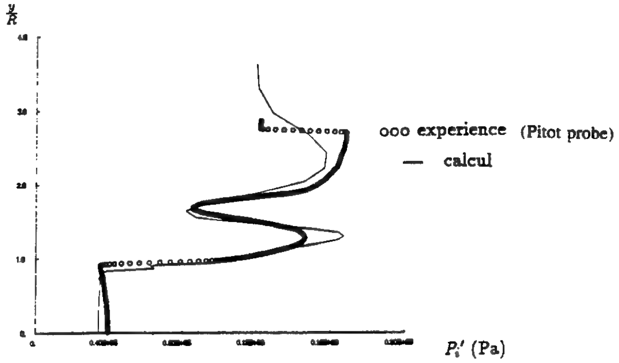


FIG. 10 - *Arrière-corps: profil de pression à x/R=6*

Calculs instationnaires

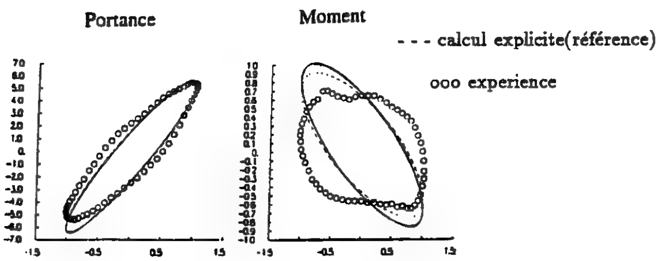


FIG. 1 - *Schéma implicite d'ordre 1 en temps (cfl=400)*

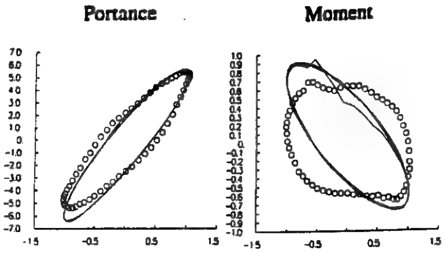


FIG. 2 - *Schéma Runge-Kutta implicite d'ordre 2 en temps (cfl=400)*

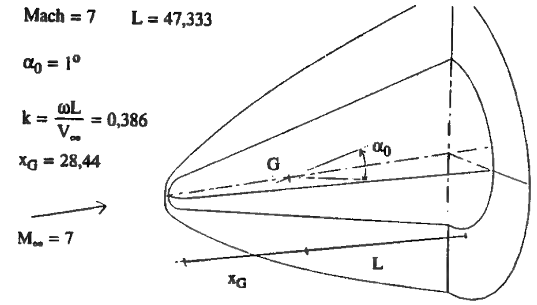


FIG. 3 - *Profil de sphère-cône*

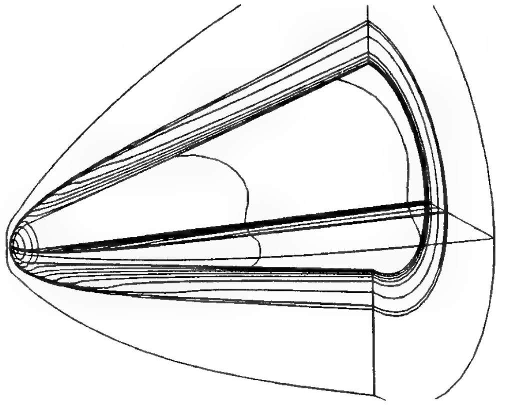


FIG. 4 - *Lignes iso-nombre de Mach à 3 cycles* $\Delta M = 0,5$

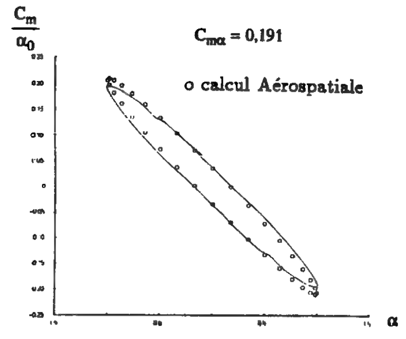


FIG. 5 - *Moment par rapport au centre de gravité G*

Calcul DDLU 3D

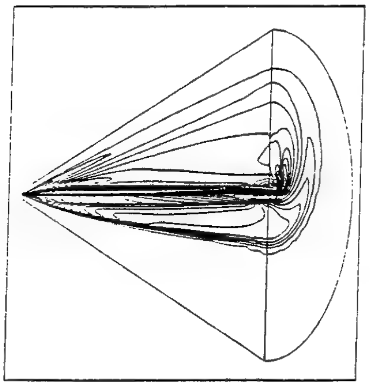


FIG. 11 - *Fuselage lenticulaire: lignes iso-nombre de Mach*

The Computation of Aircraft Store Trajectories using Hybrid (structured/unstructured) Grids.

D.J.Jones, F.Fortin, D.Hawken, G.F.Syms and Y.Sun,
Institute for Aerospace Research,
National Research Council,
Montreal Rd.,
Ottawa K1A 0R6, CANADA
E-mail: denis.jones@nrc.ca

Summary

With the high costs associated with flight and wind tunnel testing, the computation of aircraft store trajectories is becoming more important to the military establishment. In Canada, the Department of National Defence (DND) requested IAR to acquire/ develop the necessary tools to carry out the prediction of stores on release from aircraft - particularly the DND's CF-18 aircraft. After debate whether to use structured Chimera schemes or unstructured schemes, IAR decided to use the latter techniques as there was already a development program underway in that field of research. IAR had already demonstrated that hybrid (structured/ unstructured) grids had produced successful results and decided to pursue this approach for the unsteady 3-D computations. To this end, a study has been made in the 2-D case of a 'store' moving from the parent 'body'. Grid generation is underway for the full CF-18 aircraft using a commercial code and several simpler cases have been gridded and computations made in a steady 3-D environment.

1. Introduction

Accurate prediction of the trajectory of a store released from an aircraft is critical in assessing whether the store can be released safely. The trajectory of stores released in aircraft flowfields has always been difficult to predict. A typical wind-tunnel/ flight test program intended to ensure that the store will release properly is lengthy and costly. It may involve 20 flight tests, one or two wind tunnel entries, and extend over a period of several years. In the event of an improper trajectory, pylon and/or attachment point modifications may have to be made resulting in more flight and wind-tunnel testing (Ref 1).

However, with the advancement of computational fluid dynamics (CFD) techniques, a much faster prediction of carriage and trajectory data is believed to be possible. In particular, a sufficiently reliable computed flowfield data could reduce the test matrix and supplement the measured data such that the additional testing could be reduced or eliminated. Further, it is anticipated that computed flowfields could serve as a diagnostic aid in deciding among possible solutions to design problems. Both multiblock structured overlapping (Chimera - see for example [2,3]), and unstructured grid methods (for example [4] and [5]) have been used to solve multi-component and moving-body systems.

Both Chimera and unstructured methods have their advantages and disadvantages and after careful consideration IAR decided to take the unstructured grid route for its main thrust at tackling the problem, one of the main reasons being availability of codes. Also with multiblock techniques the grid cells tend to stay small and very stretched in some areas remote from the aircraft making the method less efficient. Several commercial codes were at first considered as possible contenders for predicting store release. Most of them were rejected after an initial survey and only the two codes RAMPANT [6] and FASTRAN [7] finally were on the 'short list'. Several test cases were run on the short listed codes.

After in-house evaluation of these codes it was found that neither was fully satisfactory and attention was turned to acquiring only a suitable grid generation program. Thus IAR

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

evaluated the packages I-DEAS [8], FLITE3D [9](also contains a solver) and ICEM [10] and eventually came to the conclusion that ICEM was the best package. IAR then decided to develop its own 3-D Euler solver from the existing 2-D solver which turned out to be not too time consuming. Validation of this 3-D code, called FJ3SOLV, is covered in the paper below and it will be seen that promising results are obtained. Eventually the aim is to develop this 3-D solver into a fully time dependent code with moving store and grid but in the meantime a study in 2-D has been underway and is reported in the last section below. This 2-D study will be beneficial in the 3-D development as various efficiencies will be transferred to the 3-D code.

In the first section some background material on the unstructured grid developments at IAR is reviewed. Then the 3-D grid generation and solver code acquisitions/ developments and their validations are described. Finally the unsteady 2-D code development is covered.

2. 2-D Steady State Code Developments

A fully unstructured Delaunay grid generation code was developed several years ago and is reported in Refs 11 and 12. It uses the standard Delaunay triangulation technique [13] with new points added continually at the centroids of existing triangles until a criterion of required grid density is fulfilled.
The Euler equations are solved using a cell centred finite volume technique with explicit artificial viscosity as in Ref 14 . Standard acceleration techniques such as local time stepping, enthalpy damping and implicit residual smoothing are used. Solutions of the Euler equations using these grids were obtained for several airfoils and showed good accuracy compared to standard AGARD test cases [15]. On advancing to Navier-Stokes solutions and trying to get cells very close to the surface within the boundary layer it was found that the grids became of poor quality even for wall function type of grids, for example Fig 1a. Thus it was decided that a more satisfactory grid could be obtained by using structured layers of grid near the surface followed by an unstructured grid outside these layers. The structured grid layers were generated using advancing normals with some averaging to avoid clashing of the normals, in some cases, as they advanced from the surface. An example of such grids is shown in Fig 1b for the RAE 2822 airfoil. Having generated these grids for Navier-Stokes computations, the same grids were then used for Euler results. These solutions also appeared to be very accurate as shown in Fig 2a for the RAE 2822 airfoil for a medium grid of 60 points on each of the upper and lower surfaces. Similarly a Navier-Stokes solution is shown in Fig 2b and further results were presented at the ETMA workshop in Ref 16.

Thus this hybrid approach was one which was desirable to use for stores release since accurate solutions had been obtained in the 2-D steady version of the code as mentioned above. It will be shown later that the 2-D unsteady version demonstrates good results. Although it is hoped to eventually use a 3-D hybrid grid generator, none has yet been acquired.

3. Review of 3-D Commercial Unstructured Codes.

Initially it was planned to acquire a commercial code for both the grid generation and the 3-D solver. Several possible codes were rejected after a preliminary evaluation made by calling users of these codes. Codes that only had a grid generation capability were rejected as we wanted the whole package including the solver and post processing. Finally the codes RAMPANT and FASTRAN were selected for in-house evaluation and a report on these codes has been made in Ref 17. In summary, it was found that FASTRAN could not deliver good Navier-Stokes solutions for RAE 2822 and that RAMPANT, although reasonable results were obtained in some cases, was not robust and was very slow even in the Euler mode of operation. Later the FLITE3D codes [9] were evaluated but these were found to be lacking in terms of pre- and post-processing and user support.

Having reached the point of not finding a complete package for grid generation and solver, we then had to decide whether to develop our own codes. The idea of developing a 3-D grid generator was not relished whereas the code to carry out an unstructured grid Euler solver (and later Navier-Stokes) appeared to be quite feasible. Thus IAR, with a view to first acquiring a grid generation package, contacted vendors and evaluated several unstructured grid generators including I-DEAS [8] (mainly used for structural analysis) and GFEM [18]. These codes were eventually rejected as they were either very cumbersome to use or were very slow in generating fairly simple grids.

Finally, the code ICEM [10] was evaluated and was found to be quite promising. A copy of this software was also obtained for evaluation. It has a good user interface, preprocessing and postprocessing. Its CAD software can build complicated wire frame surface models efficiently, and can take point data in PLOT3D format, and IGES files from other CAD systems. This grid generation package supports the point, line and volume sources for density control and can generate 3D unstructured meshes efficiently. The Octree method is used, which refines the grid by subdividing the tetrahedron into eight smaller tetrahedra until a satisfactory grid density has been reached. Some examples of these grids are shown later.

The idea of generating structured layers of tetrahedra near the surface will be pursued with the vendor ICEM, or IAR may develop its own capability using advancing normals as was done for 2-D. In the meantime it will be used solely in its unstructured form which may be acceptable for Euler solutions.

4. Development of a 3-D Euler Solver and Validation

Rather than trying to acquire a commercial 3-D solver, IAR decided it was more suitable to develop a 3-D Euler solver from our existing 2-D solver especially since the code would eventually have to be made into an unsteady version. To make the existing 2-D code, FJSOLV, into a 3-D version FJ3SOLV was relatively easy as in the 2-D code the logic is set up so that it is driven by edges of a triangle with the flux across the edge being computed once and added/subtracted to the total flux balance for the triangles on each side of the edge. The same principle was used for 3-D but now the edge is a 'face' of a tetrahedron. In the far field there is no vortex correction as in 2-D and the Riemann invariants alone are used.

To validate the new Euler code FJ3SOLV, we first considered the RAE 2822 airfoil spread out as a straight wing between two solid walls. A boundary condition of no normal flow was imposed at the walls so that the flow should be two-dimensional with no variation across the span. The ICEM grid generation package was used to generate a grid and two views are shown on Fig 3a. This grid was then used to generate a solution shown in Fig 3b. Note that the grid was not refined about the shock wave at about 70% chord on the upper surface and so produced a result that was quite smeared out around the shock wave. On refinement of the grid around the shock, shown on Fig 4a, a much improved shock was obtained and good two dimensionality was shown with little spanwise variation in the pressure. The accuracy for the airfoil pressure distribution is demonstrated in Fig 4b where the solution at one spanwise station (FJICEM), obtained by interpolation from nearby values, is compared to a completely 2-D solution. The latter solution was obtained with the 2-D solver FJSOLV with 30 points on each of the upper and lower surfaces (called FJDJ30 on the figure); it was also run without a vortex correction and with roughly the same far field distance as in the 3-D case. The solution using FJ3SOLV took 8 hours on the SGI Power Challenge computer and used about 299,000 grid cells.

Having achieved success with the '3-D' airfoil a more challenging case of some practical interest was next considered. The M-100 wing-body configuration (ref 19) had been used when checking the RAMPANT code. It is a good case as the experimental data is quite reliable and it has been used as a test case by Grumman for a Navier-Stokes code evaluation [20]. It was also considered to be a more realistic case for the ICEM grid generator as it has to cope with the intersecting surfaces of the wing and body. Grids obtained using ICEM are shown in Fig 5a. Results at various spanwise locations are shown on Fig 5b. Here several grids/codes are compared: FJ3SOLV using the ICEM grid (designated Fjicem on the figure), FJ3SOLV using the grid generated by RAMPANT software (Fjrampant) and lastly, the RAMPANT grid with the RAMPANT solution (Rampant). It can be seen that there is good agreement to the experimental data with the differences being typical of a non-viscous 'Euler' result compared to experiment. Also note that the present results as compared to RAMPANT results give a slight improvement while the computer time is down from 30 hours to 6 hours. The RAMPANT grid had 240,000 tetrahedra while the ICEM grid had 250,000 tetrahedra.

5. Unsteady 2-D Code Development and Validation

The existing steady state 2-D unstructured grid code FJSOLV was developed into an unsteady version so that moving stores could be simulated in 2-D and some of the problems investigated in 2-D before proceeding, at some later date, to a moving body 3-D code. Thus we can investigate such items as using a 'window' around the store to keep the grid fixed there relative to the 'store', moving the grid only within a second 'window' so that not all the grid is moved, grid refinements and grid interpolation.

For moving grids, the geometric conservation law (GCL) must be satisfied in order to be consistent. This law, which establishes the relations for the conservation of surfaces and volumes of the control cells, plays a key role in this flow simulation. If this law is violated, a misrepresentation of the convective velocities is encountered [21]. For domains bounded by moving boundaries, the mesh must follow the computational domain geometries. Usually points initially on the boundaries stay attached to those boundaries at the same relative locations, as is done here. For inside point movement two approaches are used, the first uses spring analogies [22] while the second computes velocities at the nodes by some kind of diffusive process and then evaluates the displacements as the product of velocities and time step. It has been demonstrated that the first approach is not failure proof [23]. The second approach seems more promising and is used here. The velocity of an inside point is computed as the average of the velocities of the surrounding points with the velocities of the boundaries points as limit conditions. At the first time step, the inside velocities are initialized to zero. They are then successively updated by a series of Jacobi iterations. This process gives a velocity distribution similar to that obtained by a diffusive operator. For motions of big amplitude, since the velocities of the inside points are smaller than the ones on the boundaries, the faster moving nodes will overlap the slower ones, which will require local remeshing.

The new code was first tested on a standard case [24] for the oscillating NACA0012 airfoil with $M=0.755$, $\alpha_0=2.51$, $\alpha_m=0.16^{o}$ and reduced frequency 0.0814. The grid generated for this case contained four layers of structured grid as described earlier. Some very small cells just aft of the structured zone were removed using interactive software [25] to improve grid quality. Results for this standard test case are shown in Figs 6a and 6b where CN, Cm and several Cp plots are presented. The results are consistent compared to other theories which use Euler methods (for example Ref 26) and are in fair agreement with the experimental data. It was quickly realized that the fourth order time marching scheme, as used in [26], was superior to the first order scheme that several authors are still using, both in terms of speed and also smoothness of solution. The CPU time for this case was about 36 hours on the SGI Power Challenge computer; this is quite slow mainly due to some of the cells being very small. This computation was done with a window, similar to that used in [27], around the store located at a distance of 0.03 (chord=1) from the airfoil. Within this window the grid was fixed relative to the airfoil and movement of the grid was only allowed outside of the window.

Next the code was tested on a NACA0012 falling in free air with M=0.8, α=0 and a downward velocity of 0.08 (relative to unit freestream). The trend in CN with increasing time was compared to the actual steady state result for the equivalent angle of attack. This grid was completely unstructured and the window was now fixed at about 1.4 units from the body. Fig 7a shows the initial grid and also the grid after a plunge of about 1.6 units (for a chord length of 1). This CN development is shown in Fig 7b where it can be seen that the result looks quite accurate as the normal force CN is tending asymptotically to the true steady state value.

The next computation was for a more realistic (store) type of body such as an ogive-cylinder-ogive as shown in Fig 8a with an airfoil/pylon as the parent body. For a freestream Mach number of 0.4, a reduced frequency of 0.8 and a maximum velocity of 0.064, this 'store' was moved down and up, in a cyclic manner, a distance of 0.16 units (based on an airfoil/pylon chord length of 1) to check physical consistency of the results. Figs 8b and 8c show the CN and Cm developments with time for three cycles. The results look quite physical as the lift first increases as it moves downward (seeing an upwash from the fluid), then as the gap increases the lift decreases as the 'channel' effect above the body is becoming less noticeable. When the body returns upward, the lift at first decreases as the body sees a downwash from the fluid but finally increases as the channel effect is stronger. The first window in this case was set at a distance of 0.03 from the body which basically covered only the structured layers of the grid. A second window for fixing the grid was also set around the wing/pylon. This enables good grids to be maintained near the bodies where it is felt to be necessary to achieve an accurate solution. Shown in Fig 8a is the initial grid before the store starts to move, the grid at the bottom of the store's cycle and the grid after one complete cycle. A third window was also used in this case so that the grid was only allowed to move within a distance of 4 units from the centre of the airfoil/pylon. The grid was fixed outside this window allowing for greater efficiency in grid movement. The CPU time for this case on the Power Challenge was about 5 hours.

This is the current status of the unsteady development of the program. Several more tests will be performed to check accuracy and then different schemes for moving the grid and for integrating in time will be studied. Implicit time marching schemes will be investigated so that larger time steps can be taken. These enhancements will be very useful in the future development of the 3-D version of the code.

6. Conclusions

All the pieces are now in place to complete the development of an unsteady calculation applied to the prediction of the store trajectory after release from the aircraft. A suitable 3-D grid generator has been identified in ICEM and a 3-D Euler solver has been developed in-house. To optimize the development of the 3-D unsteady version of the final code a 2-D version has first been developed and presented here. Lessons learned from this development will be incorporated into the 3-D version at a later date. The six degrees of freedom (6DOF) equations defining the motion, given the aerodynamic forces as computed from the Euler code, will be incorporated into the package to provide a complete trajectory. Another future development will be to move from the Euler formulation to a Navier-Stokes one, where structured grid layers near the surface will be especially beneficial.

References

1. Fox, J.H., Donegan, T.L., Jacocks, J.L., and Nichols, R.H. (1991): "Computed Euler Flowfield for a Transonic Aircraft with Stores", Journal of Aircraft, Vol.28, pp.389-396.

2. Lijewski L.E. and Suhs H.E.

'Time-Accurate Computational Fluid Dynamics Approach to Transonic Store Separation Trajectory Prediction'. Journal of Aircraft, Vol 31, No 4, July-Aug 1994.
3. Lijewski, L.E. and Suhs, N.E. (1992): "Chimera-Eagle Store Separation", AIAA 92-4569-CP.
4. Löhner, R. and Baum, J. (1992): "Comparison of Wing/Pylon/Store Experiment with an Euler Finite Element CFD Code", AIAA 92-4573.
5. Parikh P., Pirzadeh S. and Frink N.T. 'Unstructured Grid Solutions to a Wing/Pylon/Store Configuration'. AIAA Journal Vol 31, No 6, Nov 1994.
6. Fluent Inc., Centerra Resource Park, 10 Cavendish Court, Lebanon, NH, USA.
7. CFD Research Corporation, 3325 Triana Blvd., Huntsville, Alabama, USA
8. I-DEAS, Structural Dynamics Research Corporation, 2000 Eastman Drive, Milford, Ohio, USA.
9. FLITE3D, Computational Dynamics Research, Innovation Centre, University College, Singleton Park, Swansea, UK.
10. ICEM, CFD Engineering, 2600 ETNA Street, Berkeley, California, USA.
11. Jones, D.J., and MacLeod, B. 'Solution of the Euler Equations using Unstructured Grids'. Fourth Canadian Aeronautics and Space Institute Aerodynamics Symposium, Toronto, May 1993.
12. Fortin F. and Jones D.J. 'Solution of Compressible Inviscid and Viscous Flows around Single and Multi-Element Airfoils on Unstructured Meshes'. CFD Society of Canada Conference, June 1994.
13. Weatherill N.P. 'The Delaunay Triangulation in Computational Fluid Dynamics', Computers and Mathematics with Applications, Vol 24, No 5-6, pp 129-150, 1992.
14. Jameson A., Schmidt W. and Turkel E., 'Numerical Solution of the Euler Equations by Finite Volume Methods using Runge-Kutta Time Stepping Schemes'. AIAA Paper 81-1259, 1981.
15. Yoshihara H. (1985), 'Numerical Solution of Two-Dimensional Reference Test Cases,' in Test Cases For Inviscid Flow Field Methods, AGARD AR-211.
16. Fortin F. and Jones D.J. 'Unstructured Grid Solutions using k-ϵ with Wall Functions'.Proceedings of Workshop on Efficient Turbulence Models for Aerodynamics (Notes in Numerical Fluids), Editors: A. Dervieux, J. P. Dusage, L. J. Johnston, Vieweg, to be published.
17. Fortin F., Hawken D.F., Jones D.J., Symms G.F., 'A Comparison of Two Commercial Euler and Navier-Stokes CFD Codes', CFD Society of Canada Conference, CFD 95, June 1995.
18. GFEM, Electronic Data Systems Corporation, Unigraphics Division, 13736 Riverport Drive, Maryland Heights, Mo, USA.
19. Carr M.P., Pallister K.C. (1984), 'Pressure Distributions Measured on Research Wing M100 Mounted on an Axisymmetric Body,' in Experimental Data Base for Computer Program Assessment, AGARD AR-138.
20. Marconi, F., Siclari, M., Carpenter, G., Chow, R., 'Comparison of TLNS3D Computations with Test Data for a Transport Wing/Simple Body Configuration', AIAA Paper 94-2237, 1994.
21. Zhang H., Reggio M., Trepanier J.Y. and Camarero R. 'Discrete Form of the GCL for Moving Meshes and its Implementation in CFD Schemes'. Computers and Fluids, Vol 22, No 1, pp 9-23, 1993.
22. Batina J.T., 'Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aerodynamic Analysis', AIAA Journal, Vol. 29, No. 3, March 1991, pp. 327-333.
23. Chakravarthy S.R. and Szema K-Y. 'Computational Fluid Dynamics Capability for Internally Carried Store Separation'. Rockwell Intl Corp Report SC-71039-TR, Science Centre, Thousand Oaks, CA.
24. AGARD Compendium of Unsteady Aerodynamic Measurements', AGARD-R-702, Data Set 3, 1982.
25. Trepanier J.Y., Yang H., 'ADX: Algorithms for Adaptive Discretization based on Triangular Grids', Technical Report EPM/RT-93/3, Ecole Polytechnique de Montreal, 1993.
26. Gaitonde A.L. and Fiddes S.P. 'A three-dimensional moving mesh method for the calculation of unsteady transonic flows'. Aeronautical Journal, April 1995.
27. Singh K.P. et al. 'Dynamic Unstructured Method for Flows Past Multiple Objects in Relative Motion'. AIAA Paper 94-0058, Jan 1994.

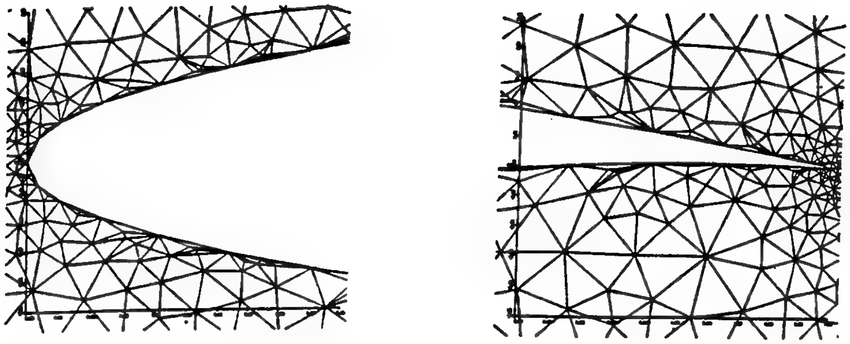


Fig 1a. Unstructured Delaunay Grid obtained for RAE 2822. Showing poor quality of grid.

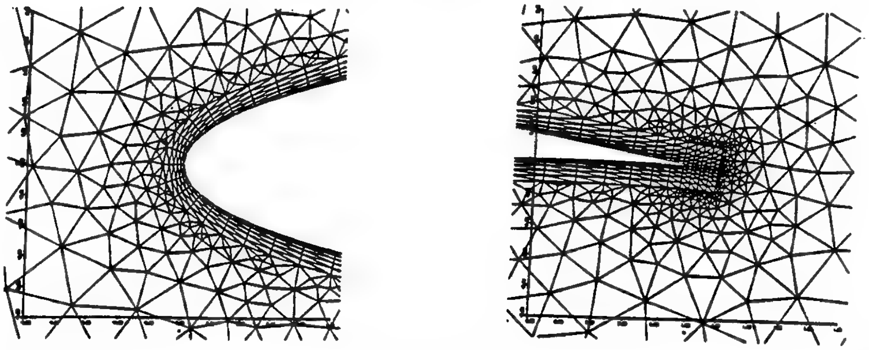


Fig 1b. Hybrid (Structured/Unstructured) Grid for RAE 2822

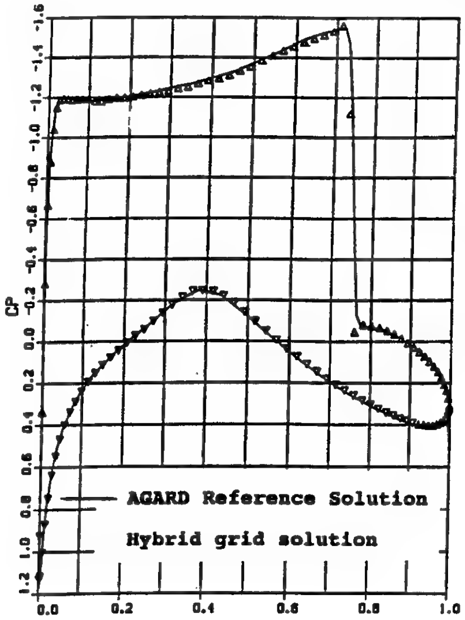


Fig 2a. Euler solution obtained with the Hybrid Grid. RAE 2822 airfoil at M=0.75, $\alpha=3^{\circ}$

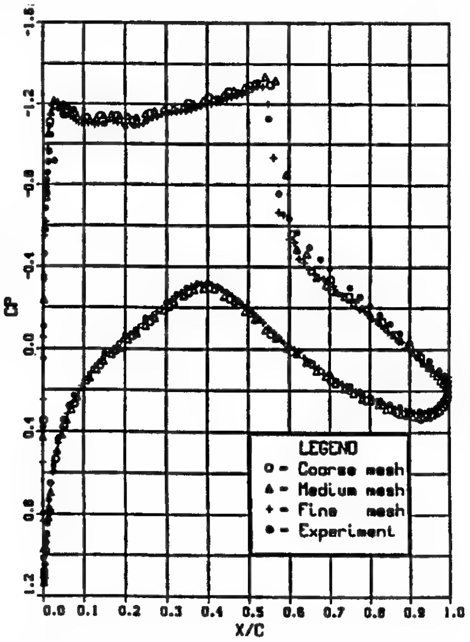
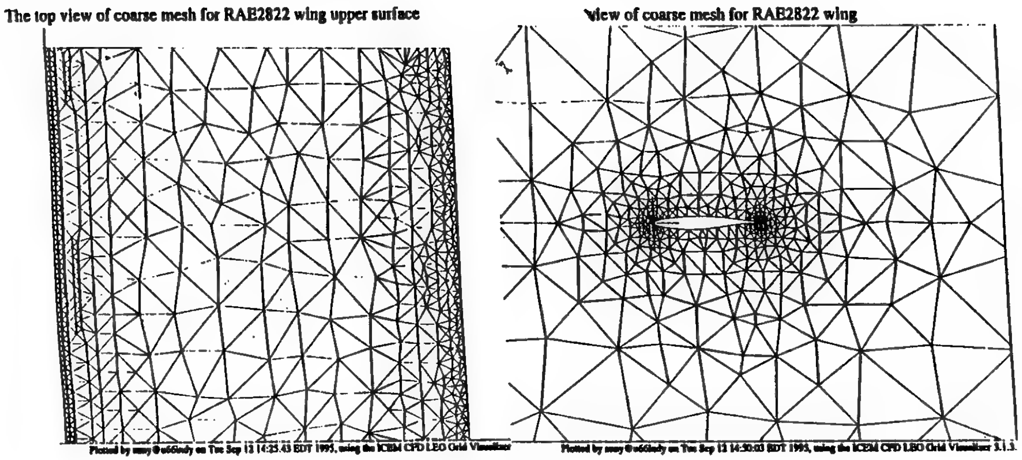


Fig 2b. Navier-Stokes Solution using Hybrid Grid. RAE 2822 airfoil, M=0.734, $\alpha=2.79^{\circ}$, Re=6.5E6


The top view of coarse mesh for RAE2822 wing upper surface
view of coarse mesh for RAE2822 wing
Plotted by suny@u66indy on Tue Sep 12 14:25:43 EDT 1995, using the ICEM CFD LEO Grid Visualizer
Plotted by suny@u66indy on Tue Sep 12 14:30:03 EDT 1995, using the ICEM CFD LEO Grid Visualizer 3.1.3.

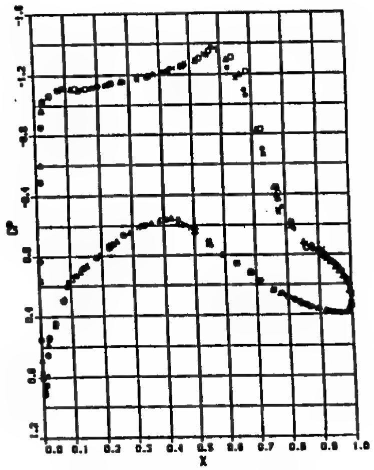
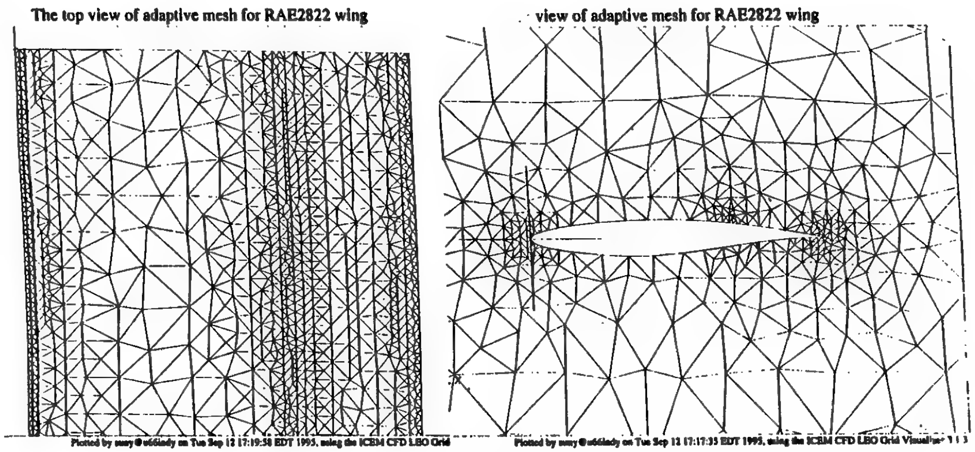


Fig 3a. Views of non-Adaptive Grid used for RAE 2822.

Fig 3b. Euler Solution obtained with the non-Adaptive Grid. RAE 2822 wing. M=0.75, α=3°, no vortex correction


The top view of adaptive mesh for RAE2822 wing
view of adaptive mesh for RAE2822 wing
Plotted by suny@u66indy on Tue Sep 12 17:19:58 EDT 1995, using the ICEM CFD LEO Grid
Plotted by suny@u66indy on Tue Sep 12 17:17:35 EDT 1995, using the ICEM CFD LEO Grid Visualizer 3.1.3

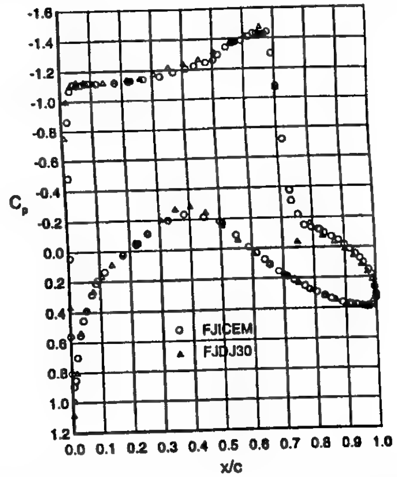
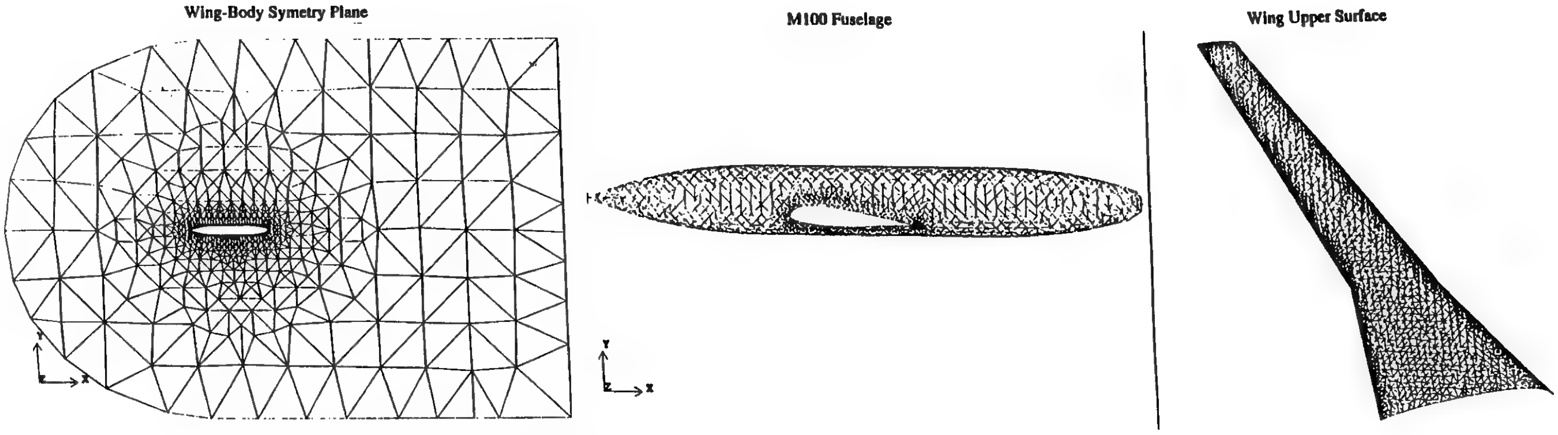


Fig 4a. Views of Adaptive Grid used for RAE 2822.

Fig 4b. Euler Solution obtained with the Adaptive Grid. RAE 2822 wing. M=0.75, α=3°, no vortex correction One spanwise section shown compared to 2-D solution

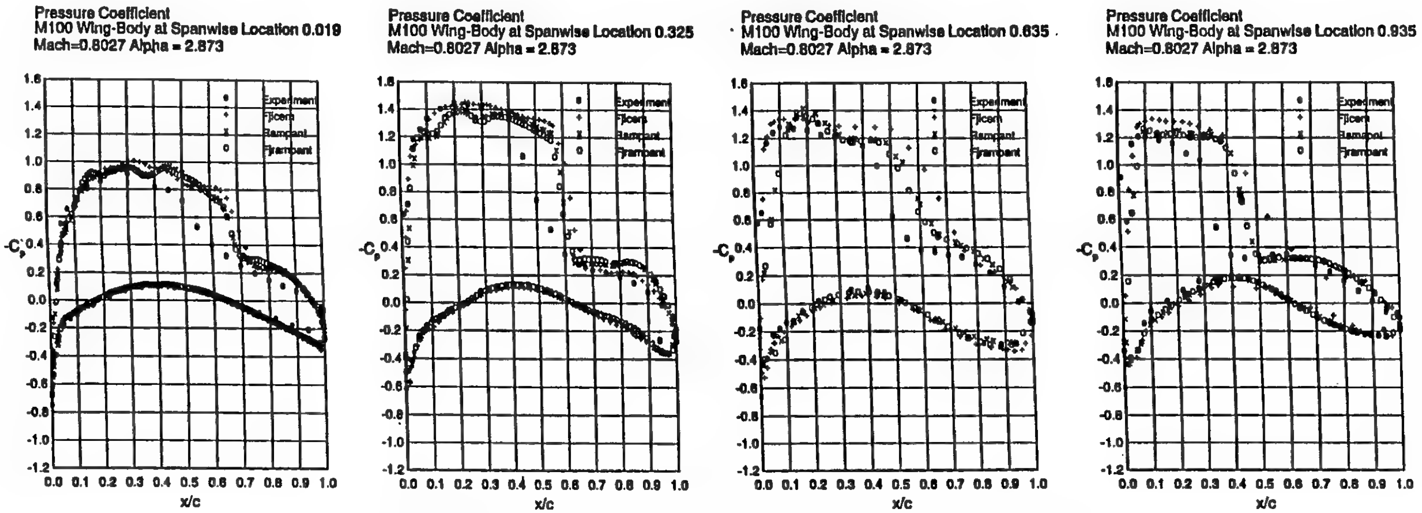


Wing-Body Symetry Plane

M100 Fuselage

Wing Upper Surface

Fig 5a. Views of the Grid for the M-100 wing-body.



Pressure Coefficient
M100 Wing-Body at Spanwise Location 0.019
Mach=0.8027 Alpha = 2.873

Pressure Coefficient
M100 Wing-Body at Spanwise Location 0.325
Mach=0.8027 Alpha = 2.873

Pressure Coefficient
M100 Wing-Body at Spanwise Location 0.635
Mach=0.8027 Alpha = 2.873

Pressure Coefficient
M100 Wing-Body at Spanwise Location 0.935
Mach=0.8027 Alpha = 2.873

Fig 5b. Euler Solutions for the M-100 Wing-Body.
Various Spanwise locations. M=0.8028, α=2.873°
Several Computations Shown.

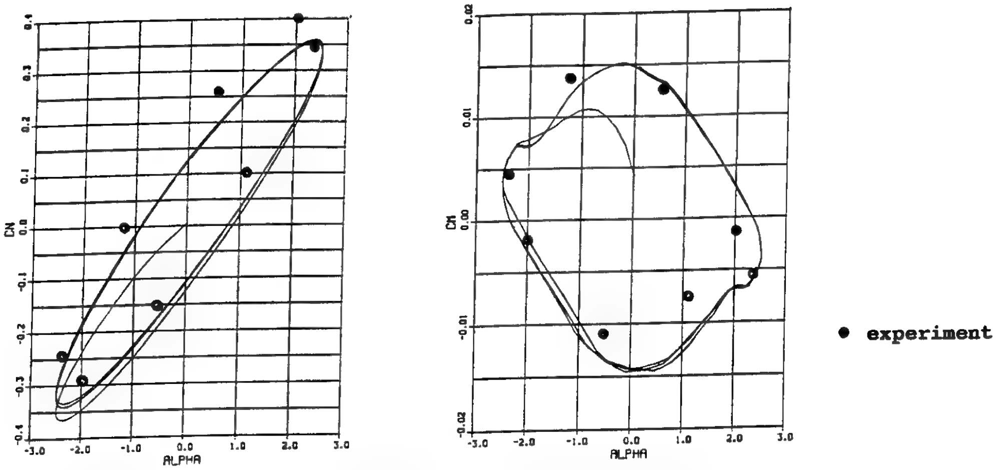


Fig 6a. Normal Force Coefficient and Pitching Moment for NACA 0012 airfoil. Showing three cycles of the motion.

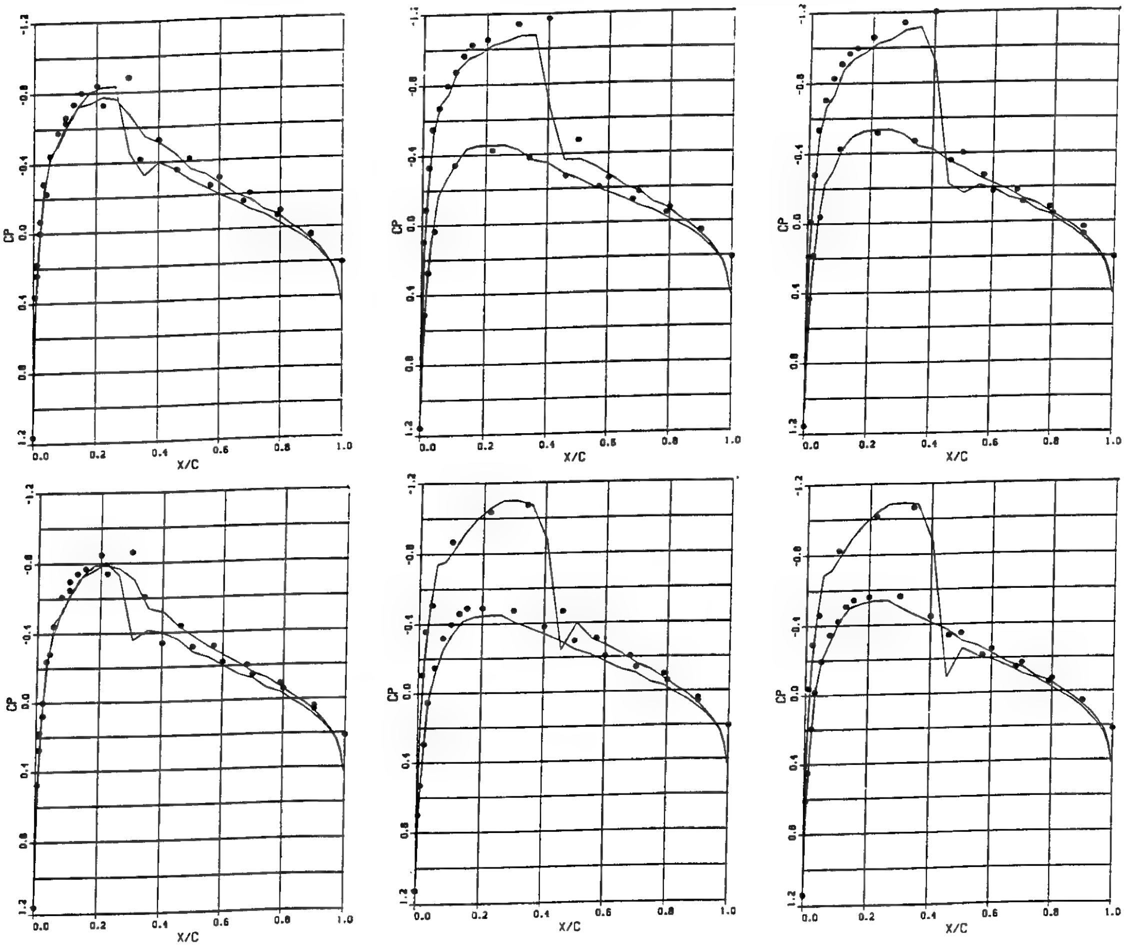


Fig 6b. Pressure distributions at Various Stages of the Motion.
α=1.09, 2.34, 0.52, −1.25, −2.41, −0.54 respectively

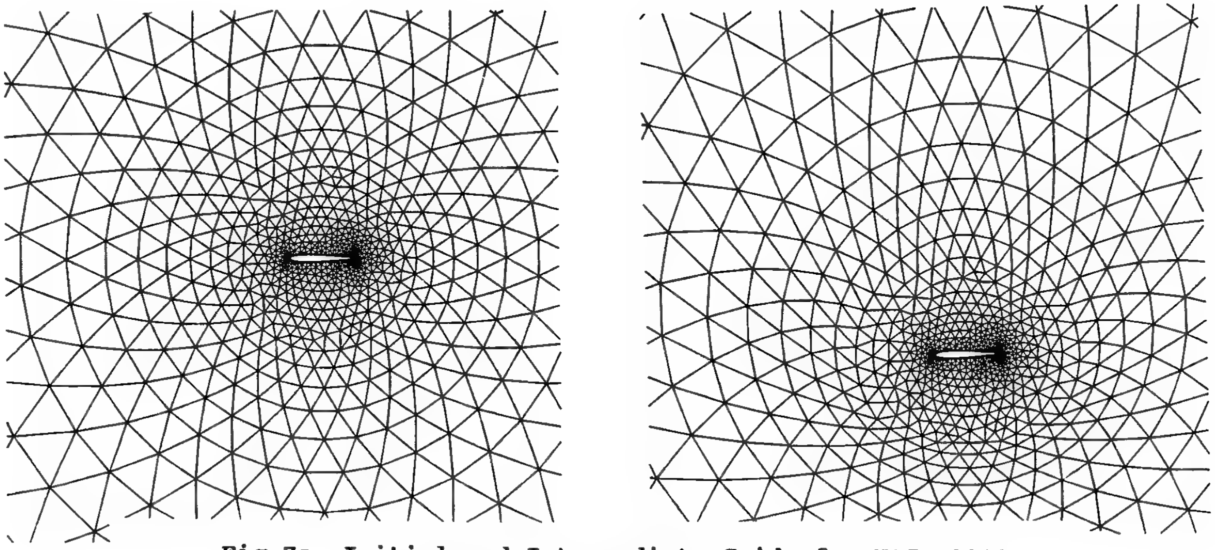


Fig 7a. Initial and Intermediate Grids for NACA 0012 in Plunging Mode

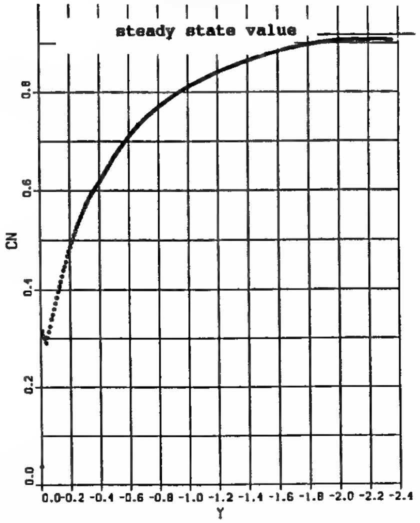


Fig 7b. Development of CN as NACA 0012 is plunging. Compared to Steady State Value

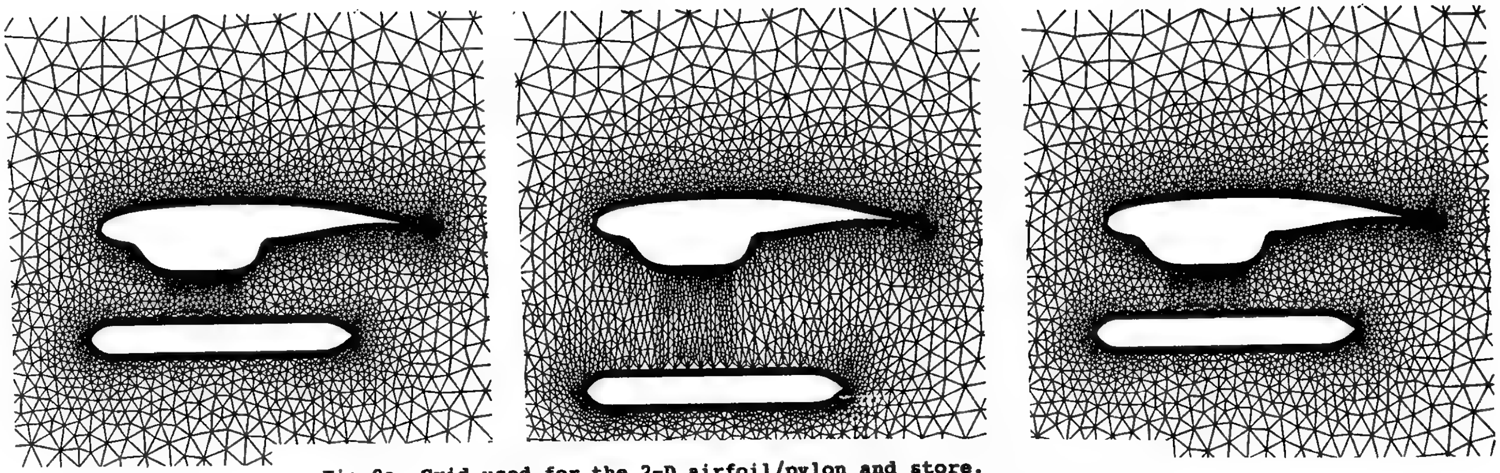


Fig 8a. Grid used for the 2-D airfoil/pylon and store.
Left: initial grid   Centre: grid at store's minimum distance
Right: grid on return to initial position

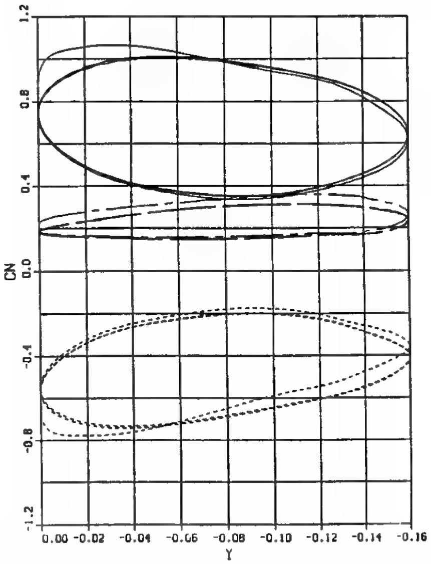


Fig 8b. CN Values for three cycles for the store (top) airfoil/pylon (bottom) and total (middle)

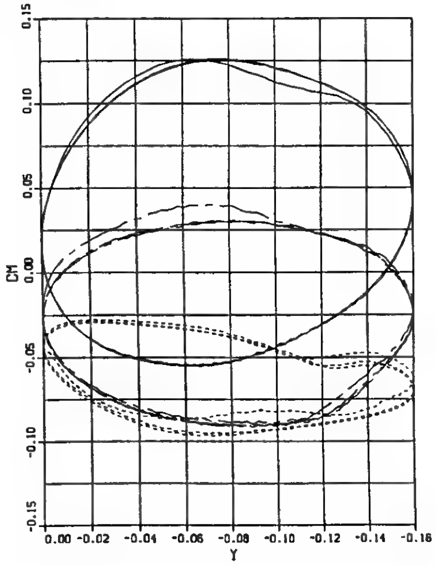


Fig 8c. Pitching Moment Values for three cycles for the store (top) airfoil/pylon (bottom) and total (middle)

Solution of the Euler- and Navier-Stokes Equations on Hybrid Grids

Martin Galle
DLR Institute of Design Aerodynamics
Lilienthalplatz 7
38108 Braunschweig
Germany

1. SUMMARY

A three dimensional finite volume scheme is presented. The scheme is based on the employment of hybrid grids, containing tetrahedral as well as prismatic cells.

The application of hybrid grids offers the possibility to combine the flexibility of tetrahedral meshes with the accuracy of regular grids. An algorithm to compute an auxiliary grid of control volumes for the entire computational domain was formulated. The dual mesh technique guarantees conservation in the entire flow field even at interfaces between prismatic and tetrahedral domains and enables the employment of an accurate upwind flow solver. Convergence to the steady state can be accelerated by a multigrid algorithm based on the agglomeration of control volumes. The formulation of such an algorithm is presented.

The code is tested on several viscous and inviscid cases for transonic and subsonic flows.

2. INTRODUCTION

The calculation of stationary flow fields around aircrafts can be regarded as one of the major tasks of CFD. Due to the progress made in the development of high performance computers, the simulation of flows even around quite complex configurations has become feasible. Therefore CFD methods nowadays have got a considerable impact on the aerodynamic design of airplanes.

One of the first requirements to be met by the applied numerical method is related to the problem turn around time. To make a scheme usable in the design process, it has to fit into industrial time scaling. Including the generation of appropriate computational grids, the solution for a certain problem should be obtained within a few days or less.

The accurate resolution of the flow field in the vicinity of solid walls has a considerable impact on the correct prediction of aerodynamic forces on the configuration. Strong solution gradients normal to the surface occur. A simulation of such flow phenomena requires a high point density in gradient direction. As for efficiency reasons usually a lower point density in the directions tangential to the wall is utilized, high aspect ratio cells are most suited for the flow resolution in those regions.

One class of schemes widely employed in practical use is based on structured grids, consisting of blocks of hexahedral cells [1]. As it is feasible to stretch hexahedral cells in one or two directions without losing grid quality, structured grids are appropriate for the simulation of high Reynolds number flows. The major drawback of structured schemes is related to the generation of suited grid for complex geometries. Grid generation normally is an iterative process [2] that requires a high level of user interaction. Though much effort has been spent in the last years to develop powerful tools, the generation of appropriate structured grids for complex geometries appears to be much more time consuming than the flow simulation.

A possibility to circumvent this bottleneck is the unstructured approach [3, 4]. The flow simulation is performed on a grid consisting of tetrahedral cells instead of hexahedra. As tetrahedral cells offer a high flexibility the discretization of complex three dimensional domains can be done almost automatically [5], with less user interaction as required for generating structured grids. The weak point of the unstructured approach is the generation of grids for high Reynolds number flows. The efficient simulation of such flows requires extremely stretched cells. The edges of tetrahedral cells of high aspect ratio are connected under very acute angles. This may cause numerical errors when the fluxes for corner nodes of such cells are evaluated. Hence, convergence and even solution accuracy can be deteriorated.

A compromise between structured and unstructured schemes is the application of hybrid schemes [6]. Hybrid grids consist of regular cells with edges exclusively normal and tangential to the surface of the geometry. In some distance from the surface, where the viscous impact on the flow has almost vanished, tetrahedral cells are employed to discretize the physical space between the regular domains and the outer boundaries. Considering the shape of the cells in the regular part there are several possibilities. The surface discretization with quadrilateral elements leads to hexahedral cells, while a surface triangulation results in prismatic cells. Due to the higher flexibility of triangles compared to quadrilaterals, a higher level of automization can be achieved when using prismatic cells.

The aim of this work is to develop of a numerical scheme that offers the possibility to employ pure tetrahedral and prismatic grids as well as hybrid grids consisting of prismatic cells in the vicinity of solid walls and of tetrahedral cells in the rest of the flow domain.

3. GOVERNING EQUATIONS

The Navier Stokes equations for the three dimensional case can be written in conservative form as

$$\frac{\partial}{\partial t}\iiint_V \vec{W}\,dV = -\iint_{\partial V} \bar{\bar{F}}\cdot\vec{n}\,dS \qquad (1)$$

where

$$\vec{W} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}$$

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

is the vector of the conserved quantities. V denotes an arbitrary control volume with the boundary ∂V and the outer normal $\vec{n}$.

The flux tensor $\bar{\bar{F}}$ is composed of the flux vectors in the three coordinate directions:

$$\bar{\bar{F}} = \vec{F}\cdot\vec{e}_x + \vec{G}\cdot\vec{e}_y + \vec{H}\cdot\vec{e}_z$$

with e_x, e_y and e_z being unit vectors in the coordinate directions. The flux vectors $\vec{F}$, $\vec{G}$ and $\vec{H}$ may be divided into its convective and viscous parts as

$$\vec{F} = \vec{F}^c + \vec{F}^v\,,\ \vec{G} = \vec{G}^c + \vec{G}^v\,,\ \vec{H} = \vec{H}^c + \vec{H}^v$$

with

$$\vec{F}^c = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho Hu \end{pmatrix}, \quad \vec{F}^v = -\begin{pmatrix} 0 \\ \sigma_x \\ \tau_{xy} \\ \tau_{xz} \\ V_x + K\frac{\partial T}{\partial x} \end{pmatrix},$$

$$\vec{G}^c = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho Hv \end{pmatrix}, \quad \vec{G}^v = -\begin{pmatrix} 0 \\ \tau_{xy} \\ \sigma_y \\ \tau_{yz} \\ V_y + K\frac{\partial T}{\partial y} \end{pmatrix},$$

$$\vec{H}^c = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho Hw \end{pmatrix}, \quad \vec{H}^v = -\begin{pmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \sigma_z \\ V_z + K\frac{\partial T}{\partial z} \end{pmatrix} \quad .$$

where

$$\begin{aligned} V_x &= u\sigma_{xx} + v\tau_{yx} + w\tau_{zx} \\ V_y &= u\tau_{xy} + v\sigma_{yy} + w\tau_{zy} \\ V_z &= u\tau_{xz} + v\tau_{yz} + w\sigma_{zz} \end{aligned}$$

The normal and tangential stresses depend on the derivatives of the velocity and on the dynamic viscosity μ:

$$\begin{aligned} \sigma_x &= 2\mu\cdot\frac{\partial u}{\partial x} - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \\ \sigma_y &= 2\mu\cdot\frac{\partial v}{\partial y} - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \\ \sigma_z &= 2\mu\cdot\frac{\partial w}{\partial z} - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \qquad (2) \\ \tau_{xy} &= \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \tau_{xz} &= \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\ \tau_{yz} &= \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \quad . \end{aligned}$$

The viscosity μ can be calculated employing the Sutherland's law:

$$\mu = \mu_\infty \cdot T^{1,5} \cdot \frac{1 + S_c}{T + S_c} \qquad (3)$$

where S_c is a constant depending on the free stream temperature $\bar{T}_\infty$:

$$S_c = \frac{110,4\mathrm{K}}{\bar{T}_\infty} \quad .$$

The pressure is calculated by the equation of state

$$p = (\kappa - 1)\rho(E - \frac{u^2 + v^2 + w^2}{2}) \quad . \qquad (4)$$

From equation (1) the temporal change of the conservative variables $\vec{W}$ can be derived as:

$$\frac{\partial}{\partial t}\vec{W} = -\frac{\iint\limits_{\partial V} \bar{\bar{F}}\cdot\vec{n}dS}{\iiint\limits_{V} dV} \qquad (5)$$

The change of the flow conditions in a certain control volume V is given by the flux over the control volume boundary ∂V related to the size of V. For a control volume fixed in time and space, equation (5) can be written as:

$$\frac{d}{dt}\vec{W} = -\frac{1}{V}\cdot\vec{Q} \qquad (6)$$

with $\vec{Q}$ representing the fluxes over the boundaries of the control volume. If the boundary is divided into n faces, $\vec{Q}$ is given by

$$\vec{Q} = \sum_{i=1}^{n}\vec{Q}_i = \sum_{i=1}^{n}(\vec{Q}_i^c + \vec{Q}_i^v)$$

where $\vec{Q}_i^c$ and $\vec{Q}_i^v$ denote the inviscid and the viscous flux over the respective face.

4. DATA STRUCTURE

The dual mesh technique is perfectly suited to be utilized in a scheme that is based on hybrid grids. From the initial grid an auxiliary grid of control volumes is generated. For a vertex based scheme, where the flow variables are stored in the nodes of the initial grid, each node is surrounded by a control volume. The boundaries of the control volumes are determined by the midpoints of cells, cell faces and edges of the initial grid. This strategy results in non overlapping auxiliary cells that fill the physical space without gaps. Figure 1 depicts such an auxiliary grid (dashed lines) for an initial hybrid grid (solid line). As it can be seen from the figure, the auxiliary grid is defined even at interfaces between the different cell types. Hence, focusing on the fluxes crossing the boundaries of the control volumes, conservativity can be guaranteed in the entire flow domain.

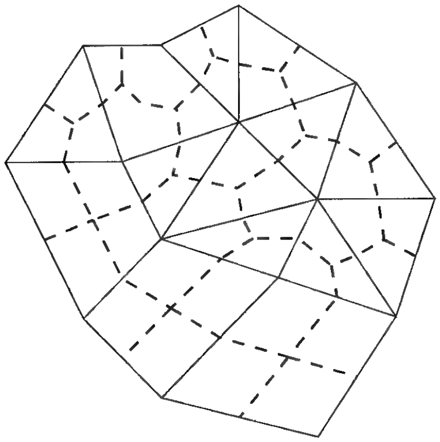


Fig. 1: Mesh of control volumes for a two dimensional hybrid grid

For each initial cell contributions to the auxiliary grid have to be determined. As this can be done cell by cell without informations about neighboring cells, the auxiliary grid can be evaluated within one loop running over all cells. Therefore, the evaluation of the auxiliary grid is quite cheap in terms of computational time.

The control volumes are composed of several faces. Each edge connecting two initial nodes is related to one face of the auxiliary grid. While in two dimensions the edges are shared by two cells, in three dimensions the number of cells sharing one edge is not constant. In figure 2 an edge connecting the nodes P_0 and P_1 is shown. It is surrounded by four tetrahedral cells. For each cell two triangles form one part of the face, so it is composed of eight triangles. As the size and the orientation of triangles can be described by normal vectors the sum of the normal vectors describes the size and the orientation of the entire face. The resulting vector is also related to the respective edge.

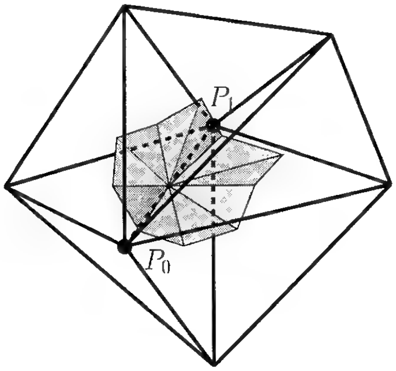


Fig. 2: Face of a three dimensional control volume

The fluxes along an edge between two nodes can be interpreted as fluxes crossing the auxiliary grid face related to the edge. The fluxes between the grid points are computed within one sweep over all edges. Informations that are required to compute the fluxes and to adjoin them to the respective nodes are:

- Geometrical coordinates of the grid nodes
- Edge to node pointer
- Components of the face vectors

Hence, both the description of the grid and the computation of the grid fluxes are based on the edges. Informations the initial grid cells are not required any more. This strategy leads to a very efficient memory allocation of less than 100 real variables per node and a good vectorization of the flow solver.

The preprocessing, including the determination of the control volumes of the auxiliary grid as well as the components of the face vectors, has to be executed before the flow calculation starts.

5. SPATIAL DISCRETIZATION

5.2 Calculation of convective Terms

The edge based data structure described in section 4 forms the basis for the employment of the accurate AUSM upwind scheme, as presented by Liou and Steffen in [7]. Considering an edge connecting two nodes P_0 and P_1, as illustrated in figure 3, the inviscid flux $\vec{Q}^c_{01}$ over Face F can be interpreted as a sum of a Mach number weighted average of the left (L) and the right (R) state at a face F and a scalar dissipative term:

$$\vec{Q}^c_{01} = |\vec{S}_{0,1}| \cdot \left(\frac{1}{2} M_F \vec{U}^+_{LR} - \frac{1}{2}\Phi_F \vec{U}^-_{LR}\right) + \vec{P} \qquad (7)$$

with

$$\vec{U}^+_{LR} = \begin{bmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a w \\ \rho a H \end{bmatrix}_L + \begin{bmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a w \\ \rho a H \end{bmatrix}_R ,$$

$$\vec{U}^-_{LR} = \begin{bmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a w \\ \rho a H \end{bmatrix}_L - \begin{bmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a w \\ \rho a H \end{bmatrix}_R$$

and

$$\vec{P} = \begin{bmatrix} 0 \\ s_x \cdot p_F \\ s_y \cdot p_F \\ s_z \cdot p_F \\ 0 \end{bmatrix}$$

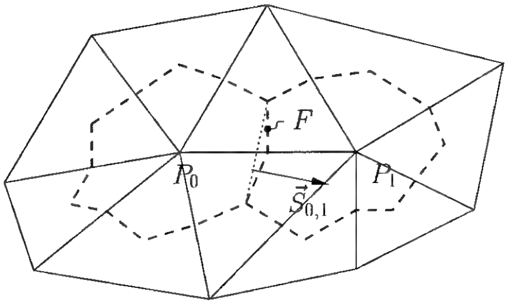


Fig. 3: Control volumes around neighboring nodes P_0 and P_1

The speed of sound a can be obtained from the relation

$$a^2 = \frac{\gamma \cdot p}{\rho} \quad . \qquad (8)$$

M_F denotes the advection Mach number at the cell face:

$$M_F = M^p_L + M^m_R \qquad (9)$$

where the split Mach numbers $M^{p/m}$ are defined as

$$M^p = \begin{cases} M & \text{if} \quad M \geq 1 \\ \frac{1}{4}(M+1)^2 & \text{if} \quad |M| < 1 \\ 0 & \text{if} \quad M \leq -1 \end{cases}$$

$$M^m = \begin{cases} 0 & \text{if} \quad M \geq 1 \\ -\frac{1}{4}(M-1)^2 & \text{if} \quad |M| < 1 \\ M & \text{if} \quad M \leq -1 \end{cases}$$

Herein M denotes the Mach number of the flow normal to the cell face.

The pressure p_F at the cell face is calculated in a similar way as

$$p_F = p^p_L + p^m_R \qquad (10)$$

where $p^{p/m}$ denote the split pressure

$$p^p = \begin{cases} p & \text{if} \quad M \geq 1 \\ \frac{1}{4}p(M+1)^2(2-M) & \text{if} \quad |M| < 1 \\ 0 & \text{if} \quad M \leq -1 \end{cases}$$

$$p^m = \begin{cases} 0 & \text{if} \quad M \geq 1 \\ -\frac{1}{4}p(M-1)^2(2+M) & \text{if} \quad |M| < 1 \\ p & \text{if} \quad M \leq -1 \end{cases}$$

The coefficient Φ_F controls the dissipation of the scheme. In the original scheme of Liou, Φ_F is set to

$$\Phi_F = |M_F| \quad . \tag{11}$$

For small Mach numbers the dissipative character vanishes since also Φ_F becomes small. In order to prevent the disappearance of the dissipation for small Mach numbers Φ_F is determined as proposed by Kroll and Radespiel in [8].

The values left and right of the face F are taken directly from P_0 and P_1 for first order calculations. For second order accurate calculations the independent flow variables are linearly reconstructed on the control volumes around P_0 and P_1. For the control volume of node P_0 it reads:

$$u_L = u_0 + \nabla\vec{u}_0 \cdot \frac{1}{2}\vec{V}_{0,1} \quad . \tag{12}$$

The gradient $\nabla\vec{u}_0$ of a variable u is obtained by employing a Green-Gaußformula:

$$\nabla\vec{u}_0 = \frac{1}{\Omega_0} \cdot \sum_{i=1}^{n} \frac{1}{2} \cdot (u_0 + u_i) \cdot \vec{S}_{0,i} \tag{13}$$

where Ω_0 is the volume of the dual cell around N_0 and $\vec{S}_{0,i}$ is the normal vector of the dual mesh face F as shown in figure 4.

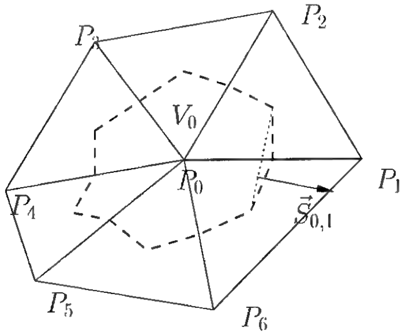


Fig. 4: Face of a three dimensional control volume

Near shocks the values on the edges have to be limited to avoid overshoots. The limiting is done by an minimum/maximum clipping like it is proposed by Barth in [9]. If a reconstructed value at any face of the control volume exceeds the minimum (or maximum) of the values given by node P_0 and the surrounding nodes $P_{1..6}$ (figure 4), the gradient $\nabla\vec{u}_0$ is scaled by a factor Θ_0, so the reconstructed value becomes equal to the minimum (or maximum) of the nodes $P_{0..6}$:

$$\nabla\vec{u}_{0,lim} = \nabla\vec{u}_0 \cdot \Theta_0 \tag{14}$$

with

$$\Theta_0 = \min(1, \Theta_1 \dots \Theta_n)$$

where

$$\Theta_i = \begin{cases} \frac{u^{min}-u_0}{u_{iL}-u_0} & \text{if} \quad u_{iL} < u_0 \\ \frac{u^{max}-u_0}{u_{iL}-u_0} & \text{if} \quad u_{iL} > u_0 \\ 1 & \text{if} \quad u_{iL} = u_0 \end{cases} \quad . \tag{15}$$

Herein u^{max} and u^{min} denote the maximum/minimum of the values of u at the nodes $P_{0..6}$ and u_{iL} denote the reconstructed value at the faces of the control volume between P_0 and P_i. The values at the faces are reconstructed as described in equation (12).

5.3 Calculation of Viscous Terms

The determination of the viscous terms is also performed edgewise. The obtained fluxes are related to the nodes associated with the respective edge. For an edge connecting the nodes P_0 and P_1 with the face vector $\vec{S}_{0,1}$ (figure 3) one obtains:

$$\vec{Q}^v_{01} = \begin{pmatrix} 0 & 0 & 0 \\ \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_y & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_z \\ V_x + K\frac{\partial T}{\partial x} & V_y + K\frac{\partial T}{\partial y} & V_z + K\frac{\partial T}{\partial z} \end{pmatrix} \cdot \vec{S}_{0,1}$$

with V_x, V_y and V_z as described in section 5.1.

The derivatives of a flow variable u have already been obtained for the second order discretization as described in section 5.2. They are the components of the gradient vectors $\nabla\vec{u} = (u_x, u_y, u_z)^T$. The face values are determined by an arithmetic averaging of the respective values in the nodes P_0 and P_1.

6. TEMPORAL DISCRETIZATION

The temporal variation of the flow quantities can be written in general form for a node P_0 as:

$$\frac{d}{dt}\vec{W}_0 + \vec{R}_0 = 0 \quad . \tag{16}$$

A comparison with equation (6) gives for the residual $\vec{R}_0$:

$$\vec{R}_0 = \frac{1}{V_0} \cdot \vec{Q}_0 \quad . \tag{17}$$

The integration in time is performed utilizing an explicit Runge-Kutta scheme, as described by Jameson in [10]:

$$\begin{array}{rcl} \vec{W}^{(0)} & = & \vec{W}(n) \\ \vec{W}^{(1)} & = & \vec{W}^{(0)} - \alpha_1 \Delta t \vec{R}^{(0)} \\ & \vdots & \\ \vec{W}_0^{(a)} & = & \vec{W}^{(0)} - \alpha_a \Delta t \vec{R}^{(a-1)} \\ \vec{W}(n+1) & = & \vec{W}^{(a)} \end{array} \quad . \tag{18}$$

Within the framework of this paper a three step scheme is employed with the coefficients

$$\alpha_1 = 0.15, \quad \alpha_2 = 0.4 \quad \text{and} \quad \alpha_3 = 1.0 \quad .$$

For the control volume surrounding node P_0 in figure 4 the convective time step Δt_0^c and the viscous time step Δt_0^v have to be determined. The resulting time step can be written as:

$$\Delta t_0 = CFL \frac{\Delta t_0^c \Delta t_0^v}{\Delta t_0^c + \Delta t_0^v} \tag{19}$$

with CFL being the Courant number. The convective time step Δt_0^c can be calculated as:

$$\Delta t_0^c = \frac{V_0}{\lambda_0^c}$$

where λ_0^c denotes the maximum eigenvalue of the flux Jacobian. It can be determined as a integration over the surface of the control volume:

$$\lambda_0^c = \sum_{i=1}^{6} |\vec{v}_{0i} \times \vec{S}_{0,i}| + a_{0i} \cdot |\vec{S}_{0,i}| \tag{20}$$

with $\vec{S}_{0,i}$ representing the face vectors of the control volume face for the ith neighbor of P_0 and $\vec{v}_{0i}$ the face velocity vector.

Following [11] the viscous time step Δt_0^v has to be scaled with a factor $K^v = 0.25$:

$$\Delta t_0^v = K^v \cdot \frac{V_0}{\lambda_0^v} \quad .$$

Employing an integration around the control volume, the viscous eigenvalue λ_0^v can be written as:

$$\lambda_0^v = \left(\frac{\gamma}{\mathrm{Pr}} + 1\right) \cdot \frac{1}{V_0} \cdot \sum_{i=1}^{6} \frac{\mu_{0i}}{\rho_{0i}} \cdot |\vec{S}_{0,i}|^2 \quad . \tag{21}$$

7. MULTIGRID ALGORITHM

7.1 Multigrid Strategies for Unstructured Grids

The acceleration of the convergence is necessary for the simulation of high Reynolds number flows. A very powerful tool that can be utilized with an explicit time-stepping scheme is the multigrid method.

Focusing on unstructured grids, there are several approaches for the formulation of a multigrid algorithm. The differences between these approaches lay in the strategy of generating the coarser grids.

One frequently utilized method employs independent grids. A set of successively coarser grids are generated around the respective geometry independently from each other [12, 13]. As the grids are not nested, expensive search algorithms are required to determine the operators needed to transfer the flow variables and the residuals between the different grids. Another drawback is the limitation of the cell size. It should not exceed the size of details of the geometry, since otherwise the correct representation of the surface can not be guaranteed.

The same holds for a different strategy, the use of telescoping points. In this case certain points are selected to remain in the coarser grids. These points are then reconnected using some triangulation algorithms [14]. The preprocessing described above has to be performed for each level again. Since one has to select existing points of the fine grid to become coarse grid points the quality of the coarse grids is worse than the quality of independently generated coarse grids. For hybrid grids this approach is not suited as different algorithms would have to be used to select and reconnect the points in the prismatic and tetrahedral regions.

The agglomeration of control volumes, as described e.g. by Lallemand et al. in [15] or Venkatakrishnan and Mavriplis in [16], can be regarded as a special case of the second approach. Certain points of the fine grid remain in the coarse grid as well, but the control volumes of the fine grid nodes are fused together in order to form the coarse grid control volumes. Since the focus is on the control volumes, the surface representation is guaranteed by definition. For a hybrid scheme using the dual mesh technique this approach is perfectly suited. As in the solution process the shape of the initial grid cells is not important in the agglomeration procedure either. One problem that may occur is the quality of the coarser grids, as one has to deal with points that exist also in the finest grid.

7.2 Agglomeration Process

The agglomeration process starts with the choice of a start node, that will remain in the coarser grid. The control volume of the start node is fused with control volumes of neighboring nodes. After having agglomerated the control volumes, the process starts again with the choice of a new start node. So agglomeration of control volumes is a greedy process, that is not expensive in terms of calculation time.

The selection of the start node and some strategy, which of the neighboring control volumes are to be fused with the control volume of the start node, is the only possibility to control the quality of the coarse grid. It appears that the best grid quality can be obtained when the agglomeration is marching along coarsening fronts throughout the grid. Furthermore, nodes lying on solid walls should be preferred to remain in the coarse grid. So the highest priority to become the next start node will be given wall nodes lying on the coarsening front.

A simple and perfectly working strategy is to fuse all control volumes of neighboring nodes that are not agglomerated yet with the control volume of the start node. Anyway, one can think about some more sophisticated algorithms, as to fuse only the n nearest neighbors, with $n \approx 8$, or, as Venkatakrishnan and Mavriplis propose in [17], to maximize the ratio of volume and surface of the coarse grid control volumes, what results in a kind of semi coarsening for Navier-Stokes grids.

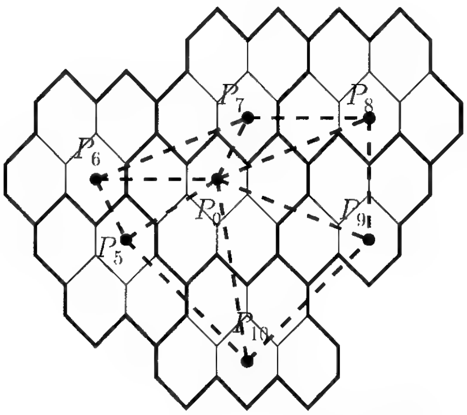


Fig. 5: Coarse grid obtained by agglomeration of control volumes

As depicted in figure 5 the coarse grid consists, as the fine grid, of a set of nodes surrounded by control volumes and connected by edges. Each edge is related to one face of the auxiliary grid of control volumes. The only difference to the fine initial grid is that the edges do not have to form any specially shaped cells any more. The informations needed to describe the coarser grid are determined directly from the fine auxiliary grid.

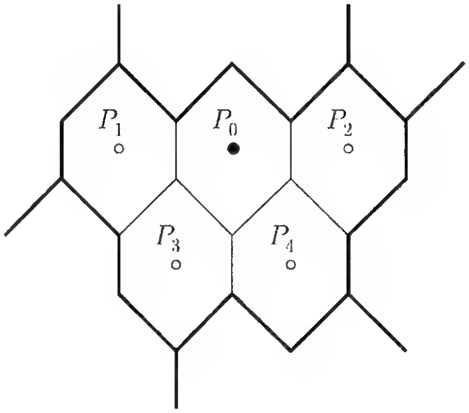


Fig. 6: Agglomeration of control volumes

Figure 6 illustrates the agglomeration of several fine control volumes to a coarse control volume around node P_0. The size of the new control volume is

$$V_{0,k} = \sum_{i=0}^{4} V_{i,k-1} \tag{22}$$

with $V_{i,k-1}$ being the size of control volume i in the grid $k-1$.

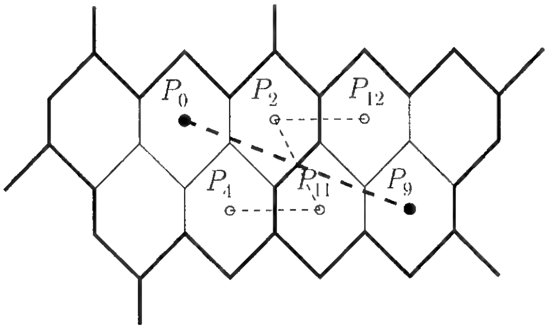


Fig. 7: Determination of coarse grid face vectors

As presented in figure 7, the normal vector $\vec{S}_{01,k}$ of the auxiliary grid face related to the edge between node $P_{0,k}$ and node $P_{9,k}$ in the coarse grid is the sum of all face vectors related to edges between children of P_0 and P_9, in this case:

$$\vec{S}_{(0,9),k} = \vec{S}_{(2,11),k-1} + \vec{S}_{(2,12),k-1} + \vec{S}_{(4,11),k-1} \quad .$$

The obtained coarse grid has got the same properties as the fine grid. Hence, the governing equations can be discretized on the coarse grid employing identical algorithms as on the fine grid. Furthermore, the coarse grid control volumes can be agglomerated again using the same strategies as for the agglomeration of the fine grid control volumes. In this way a set of grids can be created easily based on the fine grid.

7.3 Transfer of Flow Variables

The multigrid iteration starts with the performance of one time-step on the finest grid. The time step on a fine grid $k-1$ gives the solution $\vec{W}^{(a)}_{k-1}$. This solution is transferred to the next coarser grid k with a suited transfer operator $\bar{I}^{u}_{k-1,k}$:

$$\vec{W}^{(0)}_{k} = \bar{I}^{u}_{k-1,k}\vec{W}^{(a)}_{k-1} \quad .$$

As the physical position of coarse grid nodes is identical to the position of the nodes in the grid $k-1$, the transfer of flow variables from the fine grid to the coarser grid is just an injection of the respective values:

$$\vec{W}^{(0)}_{1,k} = \vec{W}^{(a)}_{1,k-1} \quad . \tag{23}$$

7.4 Evaluation of Restriction Operator

Following [10] the *Forcing-Function* $\vec{P}_k$ can be formulated as:

$$\vec{P}_k = \bar{I}^{r}_{k-1,k}\vec{R}_{k-1}(\vec{W}^{(a)}_{k-1}) - \vec{R}_k(\vec{W}^{(0)}_{k}) \tag{24}$$

with $\vec{R}_k(\vec{W}_k)$ being the residuals obtained respectively to equation (17) for the solution $\vec{W}_k$.

The restriction operator $\bar{I}^{r}_{k-1,k}$ depends on the relations between the grids $k-1$ and k. As the physical space of the coarse grid control volume around node $P_{0,k}$ in figure 6 is identical to the space of the children of $P_{0,k}$ in the finer grid, the residuals of all children have to be summed up. The contributions are weighted with respect to the size of the children cells:

$$\vec{P}_{0,k} = \frac{1}{V_{ges}} \cdot \sum_{i=1}^{4} \vec{R}_{i,k-1}(\vec{W}^{(a)}_{k-1}) \cdot V_{i,k-1} - \vec{R}_{0,k}(\vec{W}^{(0)}_{k}) \quad . \tag{25}$$

In this equation V_{ges} denotes the sum of the volumes of the children of $P_{0,k}$. According to equation (22) V_{ges} equals the coarse grid control volume $V_{0,k}$ around $P_{0,k}$, so equation (25) also be written as:

$$\vec{P}_{0,k} = \frac{1}{V_{0,k}} \cdot \left[\sum_{i=1}^{4} \vec{R}_{i,k-1}(\vec{W}^{(a)}_{k-1}) \cdot V_{i,k-1} - \vec{R}_{0,k}(\vec{W}^{(0)}_{k}) \cdot V_{0,k+1} \right] \quad .$$

As stated in equation (17) the product of residual $\vec{R}$ for any node P and the volume V of the control volume equals the fluxes $\vec{Q}$ crossing the boundary of the control volume. Therefore one can write:

$$\vec{P}_{0,k} = \frac{1}{V_{0,k}} \cdot \left[\sum_{i=1}^{4} \vec{Q}_{i,k-1}(\vec{W}^{(a)}_{k-1}) - \vec{Q}_{0,k}(\vec{W}^{(0)}_{k}) \right] \quad .$$

As the fluxes between two fine grid control volumes that are fused together cancel each other, the sum term in this equation denotes the flux over the coarse grid control volume achieved by a fine grid discretization. The forcing function can be interpreted as the difference between the fluxes obtained by a fine grid discretization and the fluxes achieved by a coarse grid discretization for a coarse grid control volume.

The temporal discretization is performed as described in section 6, while the residuals $\vec{R}_k$ are replaced by the expression $\vec{R}_k + \vec{P}_k$. Equation (18) then reads:

$$\begin{aligned} \vec{W}^{(1)}_{k} &= \vec{W}^{(0)}_{k} - \alpha_1 \Delta t(\vec{R}_k(\vec{W}^{(0)}_{k}) + \vec{P}_k) \\ &\vdots \\ \vec{W}^{(a)}_{k} &= \vec{W}^{(0)}_{k} - \alpha_a \Delta t(\vec{R}_k(\vec{W}^{(a-1)}_{k}) + \vec{P}_k) \end{aligned} \quad .$$

$\vec{W}^{(a)}_{k}$ as the basis for the next coarser grid $k+1$:

$$\vec{W}^{(0)}_{k+1} = \vec{W}^{(a)}_{k} \quad .$$

7.5 Determination of Prolongation Operator

After having determined the solution $\vec{W}^{(a)}_{1...m}$ for all grids, corrections coming from the coarse grids $k+1\ldots m$ are transferred to the grid k employing suited transfer operators. For this grid one obtains the corrected solution $\vec{W}^{(b)}_{k}$ as:

$$\vec{W}^{(b)}_{k} = \vec{W}^{(a)}_{k} + \bar{I}^{c}_{k+1,k}\vec{C}_{k+1,k} \tag{26}$$

$$= \vec{W}^{(a)}_{k} + \bar{I}^{c}_{k+1,k}(\vec{W}^{(b)}_{k+1} - \vec{W}^{(0)}_{k+1}) \quad , \tag{27}$$

while for the coarsest grid m one can write:

$$\vec{W}^{(b)}_{m} = \vec{W}^{(a)}_{m} \quad .$$

Figure 8 shows a fine auxiliary grid k and a coarser grid $k+1$. The nodes P_1 and $P_{5\ldots10}$ are coarse grid nodes. The corrections for node P_1 the correction coming from grid $k+1$ are computed as:

$$\vec{C}_{0,(k+1,k)} = \vec{W}^{(b)}_{0,k+1} - \vec{W}^{(0)}_{0,k+1} \quad . \tag{28}$$

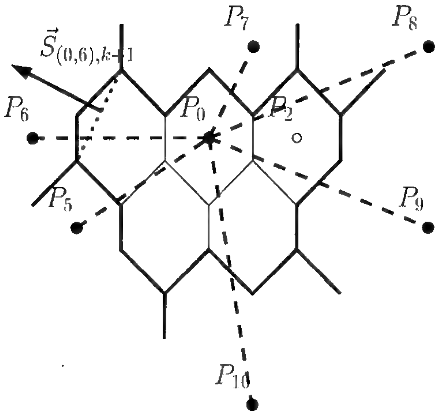


Fig. 8: Evaluation of coarse grid corrections

where $\vec{W}^{(b)}_{0,k+1}$ is the corrected value for grid $k+1$. Since the position of node P_0 is identical on both grids k and $k+1$, one can write:

$$\vec{W}^{(b)}_{0,k} = \vec{W}^{(b)}_{0,k+1} \qquad (29)$$

what leads, together with equation (23), to:

$$\vec{W}^{(b)}_{0,k} = \vec{W}^{(a)}_{0,k} + \vec{C}_{0,(k+1,k)} \quad .$$

For fine grid nodes which are also existing in the coarser grid, the corrections directly added.

The node P_2 in figure 8 does not exist in the coarser grid $k+1$. The control volume of this node has been agglomerated with the control volume around P_0. If the corrections are assumed to be constant over the coarse grid control volume, one can write:

$$\vec{W}^{(b)}_{2,k} = \vec{W}^{(a)}_{2,k} + \vec{C}_{0,(k+1,k)} \quad .$$

A higher accuracy can be obtained if the corrections are reconstructed linearly over the coarse grid control volumes. The reconstruction is similar to the reconstruction of flow variables as described in section 5.2. Using the values in the neighboring control volumes, a correction gradient $\nabla\vec{C}_{0,(k+1,k)}$ can be determined for the control volume of $P_{0,k+1}$:

$$\nabla\vec{C}_{0,(k+1,k)} = \frac{1}{V_{0,k+1}} \sum_{i=5}^{10} \vec{S}_{(1,i),k+1} \cdot \frac{1}{2}(\vec{C}_{0,(k+1,k)} + C_{i,(k+1,k)}) \quad . \qquad (30)$$

with $\vec{S}_{(1,i),k+1}$ representing the normal vector related to the edge between node P_0 and P_i. The correction in node P_2 can then be obtained as:

$$\vec{C}_{2,(k+1,k)} = \vec{C}_{0,(k+1,k)} + \nabla\vec{C}_{0,(k+1,k)} \cdot \vec{V}_{0,2} \quad .$$

where $\vec{V}_{0,2}$ denotes the vector from P_0 to P_2.

When the solution also on the finest grid is corrected, the next iteration n starts with:

$$\vec{W}^{(0)}_{1}(n) = \vec{W}^{(b)}_{1}(n-1)$$

8. NUMERICAL RESULTS

8.1 Laminar Flow over a Flat Plate

The simulation of a laminar flow over a flat plate was used to validate the formulation of the viscous term evaluation. Figure 9 depicts boundary planes of the three dimensional hybrid grid. The grid consist of three layers with 60x40 points each. The points are connected to form prismatic cells in the lower part of the grid and tetrahedral cells in the upper part.

The flow is coming from the left side parallel to the boundary planes. In rear part of the lower plane a no slip plate is located. As it can be seen from the pressure distribution on the boundary planes in figure 9 the beginning of the plate is characterized by a flow stagnation.

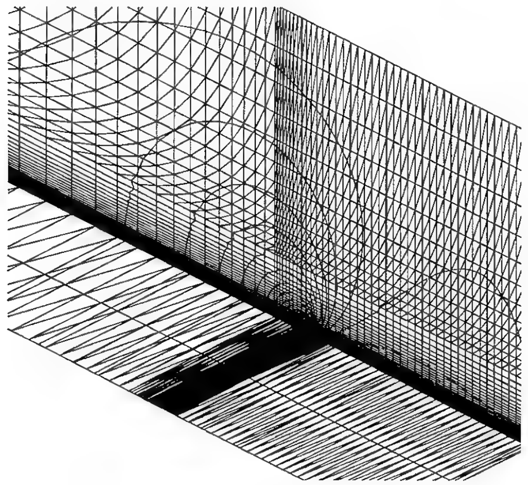


Fig. 9: Hybrid grid for flat plate and isobars

In order to create a tetrahedral grid the prismatic cells are subdivided. Hence, the point distribution is identical in both grids. Figure 10 presents the tetrahedral grid and the respective solution.

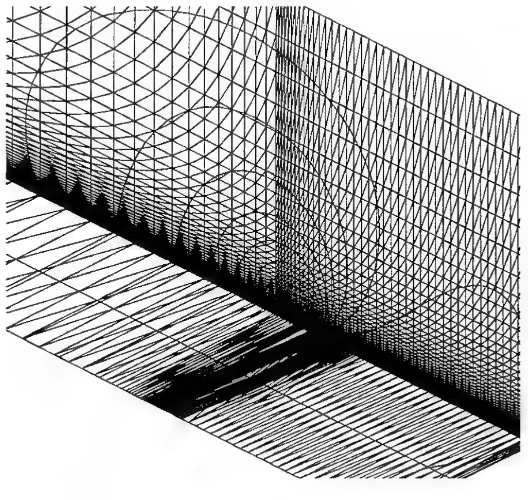


Fig. 10: Tetrahedral grid for flat plate and isobars

In figure 11 the convergence history is presented. For the calculation on the hybrid grid a convergence of 2.5 orders of magnitude is obtained after about 1600 iterations. The calculation was performed in the single grid mode in order to make it comparable to the results obtained on a pure tetrahedral grid. As one can see from figure 11 the convergence is worse for the tetrahedral case. This may be due to the disturbance of the solution caused by the diagonal edges near the wall. Furthermore, because of the higher number of edges more computational work is required for

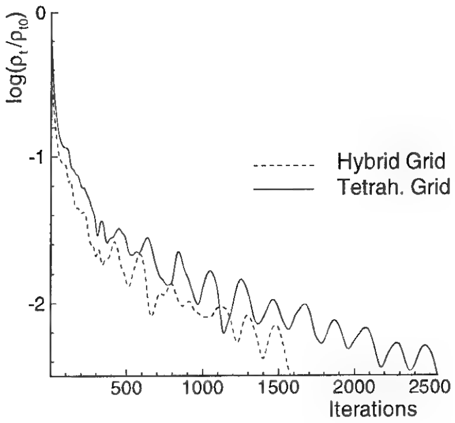


Fig. 11: Convergence history for flat plate

each time step than in the hybrid case.

Figure 12 depicts a comparison at different points between the Blasius solution and the computed solution for a subsonic ($Ma_\infty = 0.5$) laminar flow with a Reynolds Number of 5000. The solution is almost identical for both the tetrahedral and the hybrid case.

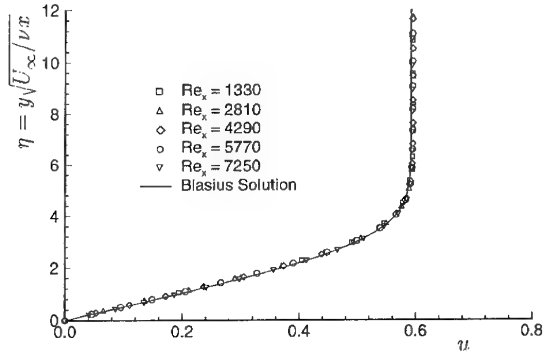


Fig. 12: Comparison of the computed solution with analytic solution

8.2 Inviscid Flow around ONERA M6 Wing

A three dimensional test case for the scheme is the inviscid flow around an ONERA M6 Wing. The prismatic part of the grid, as it can be seen from figure 13, has got an O-Topology. It consists of seven layers of prismatic cells. The triangular cell faces are located on the wing surface, while on the symmetry plane quadrilateral faces are visible. Outside the prismatic region the space is discretized with tetrahedral cells.

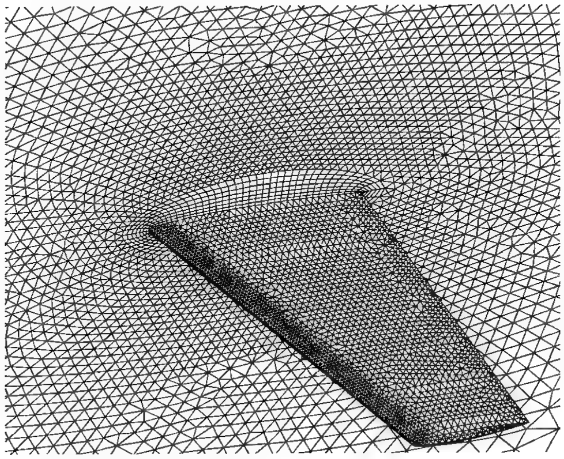


Fig. 13: Hybrid grid around ONERA M6 wing

Figure 14 depicts the solution obtained on the hybrid grid for an incidence of 3.06° and an Mach number of 0.84. The characteristic λ-shock that is visible on the upper wing surface is captured within two or three cells. No oscillations occur at the interface between the prismatic and the tetrahedral domains.

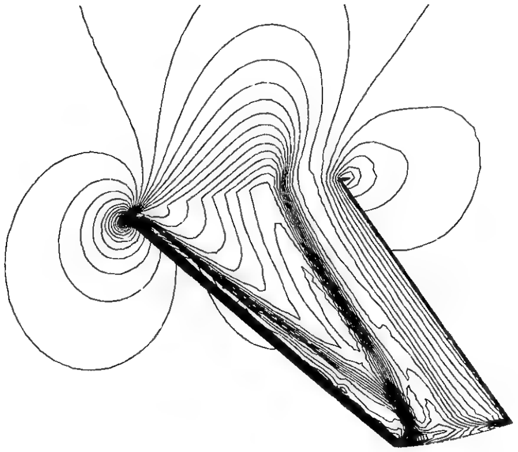


Fig. 14: Isobars of transonic flow around ONERA M6 wing

On the same grid also a subsonic flow with the free stream Mach number of $Ma_\infty = 0.5$ and an incidence of $\alpha = 3.0°$ is simulated. In figure 15 the respective solution is presented.

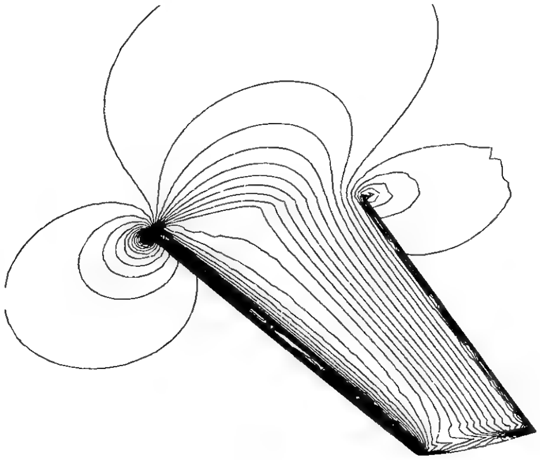


Fig. 15: Isobars of subsonic flow around ONERA M6 wing

The effect of the convergence acceleration of the multigrid algorithm is presented in figure 16. For the multigrid calculation convergence of five orders of magnitude is obtained after about 890 seconds of computational time, while on the single grid the solution has converged less than two orders of magnitude in 1500 seconds.

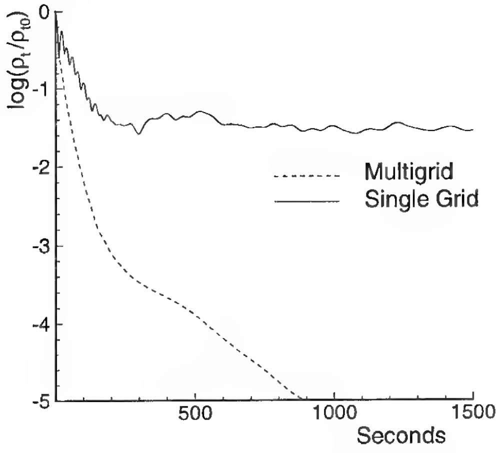


Fig. 16: Convergence history for the simulation of subsonic flow around ONERA M6 wing

9. CONCLUSIONS

A finite volume scheme based on hybrid grids is presented. The employed grids consist of prismatic cells near body surfaces and tetrahedral cells connecting the prismatic domains and the outer boundaries. The use of prismatic cells offers the possibility to resolve viscous dominated flows such as boundary layers efficiently and accurately by applying high aspect ratio cells in the respective areas. Due to the use of tetrahedral cells, grids become quite flexible and the generation of grids, even for complex configurations, is relieved considerably compared to structured approaches.

In the preprocessing an auxiliary mesh of control volumes is computed from the initial grid. The auxiliary mesh covers the entire computational domain and can be used in both the tetrahedral and the prismatic domains. In the flow solver part of the scheme an edge based data structure is utilized, so the cell structure given by the initial grid becomes unnecessary. The feasibility of employing hybrid grids even for three dimensional flow calculations are presented.

The multigrid algorithm based on the agglomeration of control volumes is a natural extension of the dual mesh technique. It fits perfectly to the edge based data structure and results in a small memory requirement.

The calculation of inviscid fluxes are demonstrated to be efficient and accurate. Shocks are captured nicely by the employed upwind flow solver. Also the formulation to calculate the viscous fluxes has proved its accuracy. At the interface between the prismatic and the tetrahedral region in some cases wiggles in the flow solution occur. Those wiggles will be subject to more detailed investigations.

Though the multigrid formulation works nicely for the case presented here, it still have to be improved, as the gain for cases with high aspect ratio cells is less than one would expect. In order to enable the simulation of viscous flows also around three dimensional geometries the next step will be the implementation of a suited turbulence model. With the improvement of the multigrid algorithm even the simulation of high Reynolds number flows are expected to become feasible for complex geometries, like flapped wings or complete aircraft configurations.

Finally a grid adaption algorithm, either based on local refinement by cell division or on global remeshing will be formulated.

10. ACKNOWLEDGMENT

The three dimensional hybrid grids that are used in this paper have been generated by J.W. van der Burg and J.E.J. Maseland of NLR within the framework of the NLR/DLR Cooperation *CFD for complete Aircraft.*

11. BIBLIOGRAPHY

[1] J.W. Sloof and W. Schmidt, editors. *Computational Aerodynamics Based on the Euler Equations.* AGARD-AG-325, 1994.

[2] A. Ronzheimer, O. Brodersen, R. Rudnik, A. Findling, and C.-C. Rossow. A new interactive tool for the management of grid generation processes around arbitry configurations. In N.P. Weatherill, P.R. Eiseman, J. Häuser, and J.F. Thompson, editors, *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, pages 441–452, 1994.

[3] A. Jameson, T.J. Baker, and N.P. Weatherill. Calculation of inviscid transonic flow over a complete aircraft. AIAA-86-0103, 1986.

[4] T.J. Barth. A 3-D upwind solver for unstructured meshes. AIAA-91-1548-CP, 1991.

[5] C. Gumbert, R. Löhner, and P. Parikh. A package for unstructured grid generation and finite element flow solvers. AIAA-89-2175, 1989.

[6] T. Minyard and Y. Kallinderis. A parallel Navier-Stokes method and grid adapter with hybrid prismatic/tetrahedral grids. AIAA-95-0222, 1995.

[7] M.S. Liou and C.J. Steffen. A new flux splitting scheme. *Computers and Fluids*, 107(1):23–39, 1993.

[8] N. Kroll and R. Radespiel. An improved flux vector split discretisation scheme for viscous flows. DLR-FB 93-53, 1993.

[9] T.J. Barth and D.C. Jesperson. The design and application of upwind schemes on unstructured meshes. AIAA-89-0366, 1989.

[10] A. Jameson. Transonic flow calculations. MAE Report 1651, Princeton University, Princeton, New Jersey, 1983.

[11] D.J. Mavriplis. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. AIAA-88-3706-CP, 1988.

[12] D.J. Mavriplis and A. Jameson. Multigrid solution of the two-dimensional Euler equations on unstructured triangular meshes. AIAA-87-0353, 1987.

[13] R.H. Bailey. A multigrid algorithm for the solution of the Navier-Stokes equations on unstructured meshes for laminar flows. DLR-IB 129-92/10, 1992.

[14] K. Riemslagh and E. Dick. Multistage Jacobi relaxation in multigrid methods for steady Euler equations. Submitted November 1993 to International Journal of Computational Fluid Dynamics.

[15] M.H. Lallemand, H. Steve, and A. Dervieux. Unstructured multigridding by volume agglomeration: current status. *Computers and Fluids*, 21:397–433, 1992.

[16] V. Venkatakrishnan and D.J. Mavriplis. Agglomeration multigrid for the three-dimensional Euler equations. AIAA-94-0069, 1994.

[17] V. Venkatakrishnan. A perspective on unstructured grid flow solvers. ICASE report 95-3, 1995.

SIMULATION DU MOUVEMENT RELATIF DE CORPS SOUMIS A UN ECOULEMENT INSTATIONNAIRE PAR UNE METHODE DE CHEVAUCHEMENT DE MAILLAGES

P. Brenner
Aérospatiale Espace & Défense
BP2 78133 Les Mureaux CEDEX, France

RÉSUMÉ

Nous présentons une méthode adaptée à la simulation numérique des **largages d'étages de fusée en présence de contraintes aérodynamiques.**
Une technique de chevauchement de maillage conservative est utilisée pour simuler le déplacement des différentes parties en mouvement dû aux efforts aérodynamiques et propulsifs.
Les équations d'**Euler en multi-gaz compressible** sont résolues au moyen d'une discrétisation du type **Volumes Finis non structurée** (tétraèdres, prismes et hexaèdres). Les caractéristiques de l'écoulement sont **localisées au centre de gravité** de chaque maille. Le schéma numérique en espace est **décentré** et du second ordre de précision sur les flux. Il est basé sur l'**algorithme de Godounov.**
La méthode d'intégration temporelle adaptative mise en oeuvre permet de simuler des écoulements fortement instationnaires avec déplacement de chocs forts tout en limitant le coût des calculs puisque les largages peuvent durer plusieurs secondes.
Le choix des algorithmes utilisés confère au code **robustesse et précision** bien que la discrétisation spatio-temporelle soit non régulière (puisque le pas de temps est différent pour chaque maille, que les maillages sont non structurés et qu'ils se chevauchent).

ABSTRACT

A computational method for the **simulation of rocket stages separations** under aerodynamical and propulsive loads is presented.
To simulate the motion of bodies, a **conservative overlapping grid technique** is used.
The flow solver is based on a **cell centred Finite Volume formulation on unstructured grids** (made of tetraedra, prisms and hexaedra).
The **Euler** equations **with mixing** gases are solved through a second order upwind scheme using the **Godunov algorithm** to compute the numerical fluxes.
To integrate equations in time, a **temporal adaptive algorithm** is used since the real duration of the simulated phenomena is long. It saves computer time and leads to accurate simulations of unsteady phenomena like acoustic waves and shocks displacements.
Despite the unregular spatio-temporal discretisation (since time steps are different in each cell, since meshes are unstructured...), the algorithms used associate accuracy with robustness.

INTRODUCTION

Les méthodes de calcul d'écoulements instationnaires autour de corps mobiles sont d'un intérêt certain pour simuler le largage d'étages de fusées vides.
En effet, les moyens d'essais susceptibles de permettre de telles simulations sont très difficiles à mettre en oeuvre:
- l'écoulement externe est fortement supersonique donc la taille de la veine de soufflerie sera faible,
- les moteurs continuent à éjecter des gaz dont la pression statique est grande par rapport à la pression externe, ce qui conduit à des éclatements de jet importants capables de provoquer le blocage de la veine,
- les gaz propulsifs sont thermodynamiquement très différents de l'air donc il faut simuler aussi cette différence de compositions,
- enfin, il faut assujettir le mouvement des corps aux forces exercées en temps réel, ce qui suppose un système complexe permettant de peser correctement ces efforts puis de les interpréter pour modifier la position des mobiles.

Ce dernier point semble très contraignant car la taille de la veine est limitée. Ainsi, il ne faut pas que le système en question soit trop encombrant et son temps de réponse doit être très bref car la durée de fonctionnement simulée est proportionnelle à l'échelle mise en oeuvre (c'est à dire qu'un largage durant une seconde en réalité durera cinq centièmes de seconde pour un moyen d'essai à l'échelle un vingtième).
Lorsque les phénomènes étudiés sont réellement instationnaires, il semble donc plus réalisable d'utiliser une approche essentiellement numérique comme nous l'avons fait avec le code FLUSEPA (réf. 7 et 8).
Le choix des formulations, des schémas numériques et des algorithmes utilisés sont consécutifs aux contraintes rencontrées pour ce type de simulations:
1- la géométrie des étages peut être complexe mais surtout, la géométrie de l'inter étage est toujours complexe (présence d'équipements, de systèmes de séparation...),

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

2- l'écoulement varie très brutalement du haut supersonique au bas subsonique et le rapport de pression rencontré dans les chocs forts peut atteindre plusieurs milliers,
3- le mouvement relatif des différents corps est totalement quelconque (rotation complexe, translation importante...),
4- l'écoulement comme la position des étages peut être rapidement évolutif et les phénomènes acoustiques ou de propagation de chocs sont souvent prépondérants quant à l'évolution du champ aérodynamique.

La première contrainte nous a amené à utiliser une formulation non structurée qui présente une grande souplesse d'un point de vue ergonomique pour la définition des maillages.

La seconde nous a conduit au choix d'une méthode de volumes finis du type Godounov (d'ordre deux) qui est très robuste et précise. Notons ;que dans la formulation utilisée, les caractéristiques de l'écoulement sont localisées au centre de gravité des mailles.

La troisième élimine toute technique utilisant les déformations de maillage. En effet, lorsque les mouvements sont importants et quelconques, le résultat d'une déformation peut conduire à une modification locale des cellules de contrôle tellement forte que le vrillage peut retourner les mailles jusqu'à l'obtention de volumes négatifs.

Quant aux méthodes utilisant des remaillages adaptatifs, elles nous ont semblé trop lourdes et contraignantes d'un point de vue instationnaire, lorsque l'on désire assurer la conservativité d'un système (ou l'accroissement entropique par exemple).

Nous avons donc opté pour une technique de recouvrement de maillage conservative. Pour tenir compte du mouvement relatif des maillages associés aux corps mobiles, nous utilisons une formulation des flux Euler-Lagrange mixtes (A-L-E) qui simplifie la méthode et surtout assure la conservativité du système. En effet, bien que les maillages soient rigides, cette technique permet de travailler dans un seul référentiel contrairement aux formulations Euleriennes pures qui nécessitent l'emploi d'un référentiel par maillage puis l'introduction de forces d'entraînement qui sont traitées comme des termes source nuisant à la conservativité globale.

Notons que pour une méthode décentrée de calcul des flux, l'A-L-E ne modifie que d'une façon mineure l'algorithme car il suffit de prendre en compte la vitesse des faces lors du décentrage. Enfin, l'interface existant entre maillages est traitée comme les faces d'une quelconque maille sans faire intervenir de changement de référentiel.

Quant à la dernière contrainte, elle nous a conduit à éliminer les méthodes d'intégration implicites qui étouffent une grande partie de l'acoustique et, au mieux, étalent les chocs qui se propagent. Nous avons donc mis au point une méthode explicite d'intégration permettant toutefois de tenir compte des caractères spécifiques de l'écoulement local: lorsque les phénomènes sont rapides dans une maille, le nombre d'itérations sur cette même maille est important, sinon, il est faible. Il s'agit d'une méthode d'intégration temporelle adaptative qui peut être considérée comme une technique consistante et conservative de pas de temps locaux.

Dans la description qui suit, nous insisterons sur les problèmes de conservativité, de consistance et de stabilité qui ont conditionné le choix des algorithmes utilisés.

MÉTHODE NUMÉRIQUE

La méthode des volumes finis (F.V.) repose sur la résolution des équations sous forme intégrale, c'est à dire que l'on fait un bilan des valeurs conservatives sur un élément de contrôle. Notons que, en toute rigueur, le bilan doit être vérifié quel que soit l'élément de contrôle considéré appartenant au domaine étudié.

1- ÉQUATIONS GÉNÉRALES

Les équations d'Euler sous forme intégrale en multi-gaz compressibles pour les écoulements tridimensionnels et lorsque l'élément de contrôle est mobile, peuvent se mettre sous la forme suivante:

$$(1)\quad \frac{\partial}{\partial t}\iiint_{\Omega}\begin{pmatrix}1\\ \rho\\ \rho\vec{V}\\ \rho Et\\ \rho Cv\\ \rho N\end{pmatrix}\partial\tau = \iint_{\partial\Omega}\begin{pmatrix}1\\ \rho\\ \rho\vec{V}\\ \rho Et\\ \rho Cv\\ \rho N\end{pmatrix}(\vec{U}-\vec{V}).\partial\vec{\sigma} - \begin{pmatrix}\vec{V}\\ \vec{0}\\ P\\ P\vec{V}\\ \vec{0}\\ \vec{0}\end{pmatrix}.\partial\vec{\sigma}$$

Où, $\delta\Omega$ est la frontière qui entoure l'élément de contrôle Ω, ρ est la masse volumique, V est la vitesse du fluide dans le référentiel Galiléen de calcul , U est la vitesse de la frontière $\delta\Omega$, E_t est l'énergie totale spécifique, Cv la chaleur spécifique à volume constant et N, le nombre de moles par unité de masse.

La première équation est relative à la variation du volume de contrôle. Puisque les maillages sont rigides, elle n'est utile qu'au niveau de l'interface entre maillages comme nous le montrerons par la suite.

Les deux équations sur Cv et N sont prises en compte pour simuler le mélange de gaz supposés parfaits et thermodynamiquement parfaits. Cette modélisation très simple n'est correcte que si le milieu est non réactif.

2- SOLVEUR AÉRODYNAMIQUE

Bien qu'il soit difficile de découpler la discrétisation spatiale de la discrétisation temporelle, nous allons procéder ainsi afin de mettre en lumière le cheminement que nous avons suivi.

2.1- Discrétisation spatiale

Calcul des flux

Approximons l'enveloppe $\delta\Omega_i$ de l'élément i par un polyèdre à *N* faces planes notées S_{ij} orientées de i vers ses voisins j.
La discrétisation la plus naturelle provient de Godounov (Réf. 1): il s'agit de considérer des états constants par morceaux sur chaque maille, alors, la méthode de calcul des flux numériques sur chaque interface consiste à résoudre le problème de Riemann ainsi posé.
Plusieurs techniques pour le résoudre d'une façon approchée ont été proposées.
A notre avis, les problèmes qu'elles suscitent tant du point de vue du manque de fiabilité (flux de Roe, d'Osher...) que de la viscosité numérique importante (flux de Van Leer...) les rendent peu attrayantes en comparaison de la méthode exacte de Godounov (Réf. n°2). Bien qu'elle ait la réputation d'être coûteuse puisque itérative, notre expérience montre que, pour un niveau comparable d'optimisation sur calculateur vectoriel (CRAY), l'algorithme de Godounov n'est au plus que de 20% plus cher que celui de Osher.
Pour toutes ces raisons, nous avons opté pour l'algorithme original de Godounov.

Notons que nous avons éliminé les méthodes dites centrées car elles nécessitent l'introduction d'un terme de viscosité artificielle paramétrable et donc ne sont généralement pas utilisables en "boîte noire".

Études la précision des approximations F.V.:

soit (2) $$\vec{F} = \begin{pmatrix} 1 \\ \rho \\ \rho\vec{V} \\ \rho Et \\ \rho Cv \\ \rho N \end{pmatrix} (\vec{U} - \vec{V})$$ continue dérivable

alors (3) $$\iint_{\partial\Omega} \vec{F}.\partial\vec{\sigma} = \iiint_{\Omega} \mathrm{div}(\vec{F})\partial\tau$$

soit (4) $\tilde{F}(M) = \vec{F}(M) + \vec{\theta}(h^n)$ une approximation du flux exact à l'ordre n en M, h étant une dimension caractéristique de Ω dont le volume est égal à ω.

Comme $\vec{F}$ est une fonction lisse on peut étudier la précision de la discrétisation au centre de gravité G de Ω lorsque l'on utilise les flux d'ordre n.
Tous calculs faits :

(5) $$\frac{1}{\omega}\iint_{\partial\Omega} \tilde{F}.\partial\vec{\sigma} = \mathrm{div}(\vec{\tilde{F}}_G) + \theta(h^2) + \theta(h^{n-1})$$

Notons que, sous certaines conditions d'approximation et de configurations géométriques particulières, le second terme de troncature gagne un ordre de précision et que, en dehors du point G, le premier terme chute à l'ordre 1.

Nous voyons donc que le schéma que nous utilisons, d'ordre 1 pour les flux est en général d'ordre zéro donc inconsistant au sens des différences finies sur les maillages quelconques et en particulier en non structuré.

Passage à l'ordre 2 en espace

Afin d'assurer la consistance du schéma, il faut donc que les états de part et d'autre des interfaces S_{ij} soient au moins calculés à l'ordre 2.
Pour ce faire, nous utilisons l'approche **M.U.S.C.L.** (Réf. 3): elle consiste à reconstruire linéairement les variables ρ, ρV, P ,Cv et N sur chaque cellule.
Alors les états nécessaires à la résolution du problème de Riemann sont du second ordre donc les flux calculés sont bien d'ordre 2.
Notre schéma d'intégration contient un point par face, il faut donc, pour préserver l'ordre de précision, que ce point soit impérativement situé au centre de gravité de la face.
Quant aux calculs des gradients, nous utilisons une méthode de moindres carrés dont le support repose sur les voisins principaux (c'est à dire ceux qui ont une face en commun avec l'élément considéré). De cette façon, nous obtenons des gradients "centrés" dont la précision est au moins du premier ordre (sur les maillages réguliers ou non), ainsi la reconstruction est bien du second ordre.

Nous utilisons pour plus de souplesse des maillages non structurés constitués de tétraèdres, de prismes, de pyramides et d'hexaèdres: ce dernier type d'éléments **permet, lorsque les maillages sont réguliers, d'obtenir une précision d'ordre 2** dans les zones "importantes", les autres types servent à faire du "remplissage". De ce fait, lorsque les faces quadrangulaires (des hexaèdres par exemple) ne sont plus planes, il faudrait deux points d'intégration par face pour assurer la consistance. Nous envisageons cette modification au schéma à court terme.

Limitation des gradients

L'approximation spatiale précédente est consistante malheureusement, son intégration en temps pose quelques difficultés:
1 - pour un schéma d'intégration Euler explicite, l'analyse de Fourier monodimensionnelle (de l'équation de transport linéarisée) démontre l'instabilité de la méthode pour les grandes longueurs d'onde (Réf. 4).

2 - quel que soit le schéma d'intégration, la méthode est fortement oscillante à proximité des zones de discontinuité (chocs, variations importantes de taille de mailles...).

La solution la plus simple au premier problème consiste à mettre en oeuvre un schéma temporel plus élaboré (schéma de Heun explicite, Euler implicite...).

Pour résoudre le second point dur, par contre, la solution que nous utilisons est apparentée à celle de Van Leer qui consiste à limiter les pentes comme suit:

Considérons pour simplifier, le cas d'une équation de convection linéaire multidimensionnelle à la vitesse $\vec{C}$ (constante) de la variable scalaire α.

soit (6) $$\frac{\partial\alpha}{\partial t}+\vec{C}.\vec{\mathrm{Grad}}(\alpha)=0$$

que l'on intègre en temps sur la maille i, par une méthode à un pas, après discrétisation du type volumes finis, sous la forme:

$$(7)\quad \alpha_i^{n+1}-\alpha_i^n=-\frac{\Delta t}{\omega}\sum_j \tilde{\alpha}_{ij}\vec{C}.\vec{S}_{ij}$$

Où Δt représente la durée comprise entre les instants n et n+1 et $\tilde{\alpha}_{ij}$ la valeur de α qui détermine le flux au centre de gravité de la face S_{ij} entre n et n+1.

On désire créer un schéma localement à variation bornée c'est à dire vérifiant la contrainte suivante:

$$(8)\quad \alpha_{\min}=\min_j(\alpha_j^n)\leq\alpha_i^{n+1}\leq\max_j(\alpha_j^n)=\alpha_{\max}$$

Nous remarquerons qu'un tel schéma est positif, et après quelques manipulations, on en déduit les conditions suffisantes mais non nécessaires:

$$(9a)\quad (\alpha_{\max}-\alpha_i^n)(\frac{1}{CFL}-1)\geq\alpha_i^n-\min_j(\tilde{\alpha}_{ij})$$

$$(9b)\quad (\alpha_i^n-\alpha_{\min})(\frac{1}{CFL}-1)\geq\max_j(\tilde{\alpha}_{ij})-\alpha_i^n$$

$$(9c)\quad \alpha_{\max}\geq\tilde{\alpha}_{ij}\geq\alpha_{\min}\forall j$$

Avec

$$(9d)\quad CFL=\frac{1}{2}\frac{\Delta t}{\omega}\sum_j\left|\vec{C}.\vec{S}_{ij}\right|$$

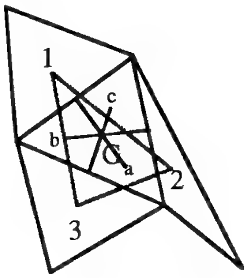
Cette formulation du CFL coïncide bien avec la formulation usuelle monodimensionnelle et avec celle de Godounov, en multidimensionnelle pour des maillages de parallélépipèdes réguliers.

Grâce à ces conditions, nous retrouvons facilement les résultats établis en monodimensionnel pour des maillages réguliers, lorsque l'on intègre par un Euler explicite:
- la méthode de Godounov du premier ordre est stable à CFL égale 1,
- le limiteur minmod est stable à CFL égale 2/3,
- le premier limiteur de Van Leer, qui correspond à la condition (9c) est stable à CFL égale 1/2...

Bien que le raisonnement précèdent soit issu de la linéarisation d'une équation scalaire, l'intérêt primordial de cet ensemble de contraintes provient du fait que:
1- il est utilisable en multidimensionnel,
2- il est applicable à de nombreux schémas en temps,
3- il n'est pas lié au type de reconstruction,
4- il est local, donc il autorise l'étude de la stabilité des méthodes à pas de temps locaux.

Dans notre code, nous utilisons le limiteur correspondant à CFL égal 1/2 en limitant globalement le gradient (sur chaque élément et pour chaque variable) en le multipliant par un coefficient qui permet de satisfaire (9a), (9b) et (9c).

Nous noterons au passage que l'emploi d'un tel limiteur peut parfois (rarement si le maillage est régulier) faire chuter la précision des flux à l'ordre 1. Par exemple pour le limiteur correspondant à CFL égal 1/2, l'ordre 2 n'est effectif que lorsque le polyèdre maximal ayant pour sommet les centres des voisins j de i contient à la fois les centres de gravité des faces S_{ji} et leur symétriques par rapport au centre de gravité G de l'élément i considéré.
Le cas bidimensionnel ci-dessous permet d'illustrer cette condition: le polygone maximal est matérialisé par le triangle (1,2,3), le centre de gravité de la face S_{G1} et son symétrique le point (a) sont bien contenus dans ce polygone, par contre, bien que le centre de gravité de S_{G3} vérifie la condition, son symétrique, le point (c) ne la vérifie pas et pour S_{G2}, c'est le contraire, seul le symétrique (b) est dans le triangle (1,2,3). Cette configuration de maillage peut donc rendre le schéma numérique inconsistant du fait de la limitation globale.



Nous n'en ferons pas la démonstration mais, nous pouvons dire que cette condition est suffisante sans être toujours nécessaire: elle dépend du champ aérodynamique étudié.

Enfin, la limitation que nous utilisons ne garantit pas la monotonie du schéma en monodimensionnel. Lorsque l'on étudie par exemple le cas d'une détente dans le vide, cette caractéristique peut devenir pénalisante: il est alors difficile sans passer à l'ordre 1 de conserver des CFL corrects. Pour pallier ce problème, nous avons mis au point une procédure "monotonisante" qui consiste à prendre la valeur la plus proche de α_i parmi $\tilde{\alpha}_{ij}$ et $\tilde{\alpha}_{ji}$ (donc la valeur la plus proche du premier ordre) et nous procédons de même du coté j. Nous voyons bien que de cette façon, la reconstruction devient monotone et de plus, sur un maillage régulier, cette correction n'est que du troisième ordre pour les flux donc le résultat global reste du second ordre lorsque le limiteur global n'est pas effectif.

2.2- Discrétisation temporelle

Schéma d'intégration en pas de temps global

Nous avons précédemment souligné le fait que le schéma du second ordre non limité est instable lorsque l'on intègre en temps par une méthode de Euler explicite. Pour remédier à cet inconvénient et afin d'augmenter la stabilité du schéma limité (donc travailler avec un CFL supérieur à 1/2), nous avons implémenté un schéma d'intégration explicite du second ordre.

En effet, l'étude de la stabilité linéaire par **l'analyse de Fourier montre que les schémas classiques d'intégration du second ordre sont stables pour un CFL égal à 1, lorsque l'on utilise une discrétisation spatiale décentrée du second ordre avec pente centrée.**

Pour obtenir un schéma relativement bon marché, il faut tenir compte du coût informatique des différents algorithmes intervenant lors de la discrétisation spatiale. Ainsi, approximativement, la résolution du problème de Riemann représente 15% du coût total, le calcul des gradients 30%, la limitation globale environ 30% et la limitation locale environ 5%, le reste étant difficilement répertoriable

Nous voyons donc qu'il faut éviter si possible de recalculer les gradients et de les relimiter globalement. Par contre, il est acceptable de refaire une limitation locale et de recalculer les flux.

Il faut, d'autre part, prendre en considération le fait que les équations à résoudre ne sont pas linéaires: tous les schémas linéarisés sont équivalents pour l'analyse de Fourier. Il faut mettre en oeuvre le schéma dont le comportement non linéaire est le meilleur.

Nous avons donc choisi le **schéma de Heun** mais avec un seul calcul de gradient. Concernant la limitation globale, une procédure simplifiée permet un regroupement avec la limitation locale sans perte de rendement.

Nous rappelons que le schéma de Heun consiste à calculer les flux (et les gradients) à l'instant n, puis grâce à cette approximation, on calcule l'état à l'instant n+1 donc les flux à l'instant n+1. Alors, la variation entre l'instant n et n+1 correspond à la demie somme des flux précédemment calculés en n et n+1.

Ce schéma est très stable pour les phénomènes non linéaires.

Schéma d'intégration temporel adaptatif

Le schéma de Heun possède de nombreuses qualités numériques, malheureusement, comme tout schéma explicite, il est pratiquement inutilisable lorsque la durée à simuler est importante.

Si l'on analyse les phénomènes intervenant lors d'une séparation d'étages, on remarque immédiatement qu'ils sont quasi-stationnaires sur presque tout le champ de calcul. Seules, quelques régions sont balayées par des courants fondamentalement instationnaires. Donc, il est intéressant, dans ces zones, d'utiliser de petits pas de temps, par contre, ailleurs, de grands pas de temps sont suffisants.

Nous avons donc mis au point une technique de pas de temps local qui est conservative, consistante et stable: l'intégration temporelle adaptative (Réf. 5 et 8).

Dans chaque maille, on travaille en utilisant le pas de temps le plus proche possible du pas de temps explicite maximum admissible.

Soit Δt_{min} le plus petit pas de temps sur tout le domaine, pour simplifier la gestion des différentes classes temporelles, on ordonne les pas de temps en puissance de 2, proportionnellement à Δt_{min}..

C'est à dire que si le pas de temps admissible dans la maille i vaut Dt_i alors on le transformera en:

$$(10) \qquad \Delta t_i = \Delta t_{min}.2^{Li}$$

où Li représente le niveau temporel de la cellule i tel que:

$$(11) \qquad \Delta t_{min}.2^{Li} \leq Dt_i < \Delta t_{min}.2^{Li+1}$$

Entre deux mailles, on posera comme principe que l'interface est du niveau temporaire le plus fort.

Pour que la méthode soit conservative, il faut que les intégrales de flux de part et d'autre de l'interface S_{ij} soient identiques. Donc, il suffit de définir en tout instant le flux de façon univoque sur S_{ij}. Ensuite, si par exemple Li=Lj+1 alors, dans la maille j on fera deux itérations pour une seule dans i. Ainsi dans j l'intégrale de flux vaudra:

$$(12) \qquad \int_{t}^{t+\Delta t_j} \vec{F}.\vec{S}_{ji} \partial t + \int_{t+\Delta t_j}^{t+2.\Delta t_j} \vec{F}.\vec{S}_{ji} \partial t = \int_{t}^{t+2.\Delta t_j} \vec{F}.\vec{S}_{ji} \partial t$$

et dans i:

$$(13)\quad \int_{t}^{t+\Delta t_i} \vec{F}.\vec{S}_{ij}\partial t = -\int_{t}^{t+2.\Delta t_j} \vec{F}.\vec{S}_{ji}\partial t$$

Ce qui prouve la conservativité du système.
Soit à résoudre:

$$(14)\quad \frac{\partial}{\partial t}\iiint_{\Omega} \alpha + \iint_{\partial\Omega} \vec{F}.\partial\vec{\sigma} = 0$$

qui, sur les solutions lisses, équivaut à:

$$(15)\quad \frac{\partial \alpha}{\partial t} + \mathrm{div}(\vec{F}) = 0$$

Supposons maintenant une approximation du flux telle que:

$$(16)\quad \tilde{F}(g,t) = \vec{F}(g,t) + \theta(\Delta t^{m}) + \theta(h^{n}) + \theta(\Delta t^{P}.h)$$

où g représente le centre de gravité des faces S_{ij}.
Alors, si l'on intègre ce flux sur la bordure de l'élément i en considérant que les troncatures spatio-temporelles sont indépendantes pour toutes les faces, on peut étudier la précision de l'approximation à l'instant Tm (milieu des deux bornes temporelles d'intégration) et en G (centre de gravité de Ω):

(17)

$$\frac{\partial \alpha_G^{Tm}}{\partial t} + \mathrm{div}(\vec{F}_G^{Tm}) = \theta(h^{n-1}) + \theta(\frac{\Delta t^{m}}{h}) + \theta(\Delta t^{p})$$
$$+\theta(h^{2}) + \theta(\Delta t^{2})$$

Pour que le schéma reste consistant, il faut que m soit supérieur à 1 donc le schéma doit être au minimum d'ordre 2 en temps sur les flux (et s'il n'est que d'ordre 2, une condition du type Δt/h borné est nécessaire)
Le terme en Δt^{P} correspond à l'approximation temporelle des gradients: dans le schéma de Heun que nous utilisons, nous calculons les gradients une seule fois, au début de chaque itération, donc p égale 1 et **le schéma est globalement du premier ordre en temps** (résultat déjà acquis puisque m égale 2).
Quant à la condition Δt/h borné, elle est automatiquement remplie par celle de CFL.

L'étude de stabilité des schémas temporels adaptatifs est relativement difficile puisque l'analyse de Fourier n'est plus utilisable. On peut par contre étudier la diffusivité du schéma pour l'équation la plus simple sur un maillage régulier:

$$(18)\quad \frac{\partial \alpha}{\partial t} + \frac{\partial \alpha}{\partial x} = 0$$

Si le schéma est diffusif, il possède des chances d'être stable, sinon, il est instable.

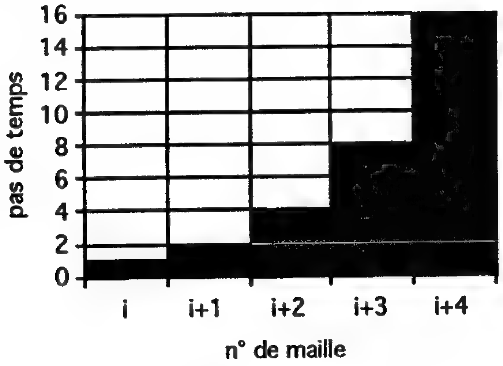
Pour un schéma de Heun du premier ordre en espace, la condition de positivité de la diffusion numérique est simple puisqu'il suffit que le plus grand pas de temps vérifie la condition de CFL. Malheureusement, pour un schéma d'ordre 2 en espace, la condition dépend de la méthode mise en oeuvre pour gérer l'ensemble des mailles.
Une approche plus facile consiste à utiliser les contraintes permettant au schéma d'être à variation bornée.
Commençons par limiter le pas de temps sur le voisinage tel que

$$(19)\quad Li = \min_{j}(Lj)$$

puis, réduisons le saut de pas de temps entre mailles tel que:

$$(20)\quad |Li - Lj| \le 1, \forall(i,j)$$

Cette dernière opération nécessite L_{max}-1 itérations L_{max} étant le niveau temporel maximum.
Nous obtenons ainsi le type de configuration suivante



en monodimensionnel.
Ainsi la maille i+4 fera une itération pour seize itérations de la maille i.
Un raisonnement simple nous montre que le schéma reste localement à variation bornée lorsque l'on limite les gradients au début de l'itération puis à la fin en conservant le α_{min} et le α_{max} calculé au début de l'itération. De plus, si la condition $Li = \min(Lj)$ n'est plus respectée, le schéma n'est plus à variation bornée.
En pratique, nous avons remarqué que la méthode est stable pour un CFL égal à 1 alors que le limiteur ne garantit la stabilité que pour un CFL de 1/2.
Réduire le saut de niveau temporaire est a priori inutile du point de vue de la stabilité mais cette procédure simplifie énormément la gestion des mailles (en particulier lors du calcul des gradients).

Concernant l'efficacité de la méthode (qui est définie comme étant le rapport: coût du calcul en pas de temps global sur coût de calcul en temporel adaptatif), on peut l'évaluer simplement lorsque l'on connaît la fonction de répartition des pas de temps. Dans la pratique, cette fonction dépend du maillage et des phénomènes locaux. Elle est donc variable. Le tableau ci dessous détermine les limites de cette efficacité ainsi qu'une valeur moyenne et une valeur expérimentale (la valeur moyenne correspond à une répartition équiprobable des pas de temps)

$\frac{\Delta t_{max}}{\Delta t_{min}}$	2	4	8	16	32	64
Ef_{max}	1,6	3,2	6,4	12,8	25,6	51,2
Ef_{moy}	1,25	2,05	3,5	6,05	10,65	19,1
Ef_{exp}					11,2	20

Notons que ces valeurs sont intimement liées à l'algorithmique utilisé, et que les formulations non structurées sont bien adaptées à ce type de technique.

3- CHEVAUCHEMENT DE MAILLAGES

L'idée de base est très simple (Réf. 6 et 7): considérons qu'un maillage M1 se comporte comme un masque qui se déplace et recouvre partiellement un autre maillage M2. L'interface entre maillages est créée naturellement: il s'agit de la surface d'intersection formée par l'évidement dans le second maillage M2 du volume occupé par le masque M1.
Les mailles de M1 ne subissent donc aucune modification. Par contre, dans M2, il y a présence de trois types de mailles (cf. figure 1):
1- les mailles totalement couvertes,
2- les mailles partiellement couvertes,
3- les mailles totalement découvertes.
Les mailles de la seconde catégorie sont donc modifiées car une partie des faces qui les constituent est couverte et de nouvelles faces correspondant à la limite externe du masque sont créées. Ces nouvelles faces forment l'interface entre mailles de M2 (coupées) et mailles de M1 (non modifiées).

3.1- Calcul des intersections

Pour simplifier le problème géométrique nous avons considéré que toutes les faces des éléments sont planes. Les faces quadrangulaires sont donc traitées comme deux faces triangulaires. Nous n'avons donc plus qu'un seul type de facette: le triangle.
L'algorithme de calcul d'intersection se limite à deux étapes:
1- déterminer le niveau de couverture de chaque face du maillage par le masque,
2- déterminer la partie de la limite externe du masque qui ferme chacune des mailles de M2 coupées.
Les deux étapes sont en fait identiques d'un point de vue algorithmique lorsqu'on les reformule comme suit:

1- Pour chaque face de M2, déterminer la partie comprise dans chacune des mailles de M1. La somme de ces parties détermine la couverture des faces.
2- Pour chaque face formant la frontière de M1, déterminer la partie comprise dans chacune des mailles coupées (une maille étant considérée comme coupée si l'une de ses faces est partiellement ou totalement couverte <u>et</u> si au moins une de ses faces n'est pas totalement couverte).
Nous voyons donc qu'il s'agit bien du même algorithme de base: déterminer la surface d'un triangle contenue dans un polyèdre.
Pour résoudre ce problème, le plus simple est de travailler dans le plan de la face triangulaire. On détermine alors la trace polygonale du polyèdre dans ce plan (chose simple puisque le polyèdre est formé de triangles) puis la partie commune au triangle et à ce polygone. Cette dernière opération nécessite juste la connaissance des segments orientés qui constituent la trace. Il faut éviter tout algorithme qui détermine le chaînage des segments entre eux: c'est inutile et excessivement coûteux.
Calculer la surface couverte n'est pas suffisant, il faut aussi déterminer son centre de gravité. On détermine alors le centre de gravité des interfaces coupées de M2 et celui des morceaux de la bordure de M1 qui ferment les mailles coupées.
Les volumes et les centres de gravité des mailles coupées de M1 sont alors calculés en utilisant les formules suivantes (Green):

$$(21)\quad \omega_i = \frac{1}{3}\sum_j \vec{OM}_{ij}.\vec{S}_{ij}$$

$$(22)\quad \vec{OG} = \frac{1}{4\omega_i}\sum_j \vec{Og}_{ij}(\vec{OM}_{ij}.\vec{S}_{ij})$$

où:
- M_{ij} est un point quelconque de la face plane S_{ij} orientée vers l'extérieur de i,
- j est soit un voisin "naturel" (donc une autre maille de M2), soit recouvrant (donc une maille de M1 qui est voisine par l'interface M1-M2),
- G étant le centre de gravité de i et g_{ij}, le centre de gravité de S_{ij}.

3.2- Optimisation du nombre d'opérations

Pour que la méthode soit utilisable, il faut que le temps de calcul des intersections soit au plus du même ordre de grandeur que le temps de calcul d'une itération de solveur aérodynamique.
Soit N le nombre de mailles, alors, le nombre de facettes à la bordure de M1 est de l'ordre de $N^{2/3}$ et le nombre de cellules coupées est du même ordre.
Pour déterminer la surface de la bordure qui ferme chaque cellule coupée, il faudra, <u>par cellule</u>, environ $N^{2/3}$ opérations.
Pour toutes les cellules, il faudra donc de l'ordre de $N^{4/3}$ opérations. Le solveur aérodynamique nécessite

de l'ordre de N opérations par itération. Il faut donc optimiser le calcul des intersections.

La solution que nous avons retenue consiste à déterminer, sur une grille cartésienne (i,j,k) contenant N mailles, l'appartenance des différentes facettes formant la bordure de M1. Ensuite, pour chaque cellule coupée, on détermine sa position dans la grille et donc quelles facettes peuvent la fermer. Le préconditionnement des facettes nécessite environ N opération, et le calcul par cellule coupée est de l'ordre de une opération soit, pour toutes les cellules environ $N^{2/3}$ opérations.

Le coût global est de N opérations (le préconditionnement est plus cher que le calcul d'intersection !) donc compatible avec l'aérodynamique.

3.3- Priorité de maillages

L'utilisation d'un masque et d'un maillage masqué manque de souplesse. En effet, les mailles du maillage masqué sont par exemple mieux adaptées au calcul d'une couche limite autour du corps lié à ce maillage que les mailles du masque.

Il est donc intéressant de définir des zones prioritaires que le masque ne peut couvrir mais qui au contraire, couvrent le masque.

La figure 2 nous montre le résultat d'une telle stratégie. Sa mise en oeuvre ne pose pas de problème particulier.

3.4- Calcul des flux à la frontière et assemblage

Les flux à la frontière sont calculés de la même façon que les flux entre deux mailles appartenant au même maillage: puisque nous travaillons en non structuré, la topologie importe peu donc une interface entre mailles sera traitée toujours de la même façon, que ces mailles appartiennent ou non au même maillage.

Quant à la première équation du système (1) concernant la variation de volume, elle permet, lorsque les mouvements sont lents, d'éviter de recalculer les intersections après chaque itération aérodynamique:`

- pour chaque maille coupée i, on évalue l'incrément de volume $\Delta\omega_i$ dû au mouvement relatif des maillages,
- on détermine l'incrément relatif maximum sur toutes ces mailles

$$(23)\quad \Delta I_{max} = \max_i\left(\left|\Delta\omega_i\right| / \omega_i^0\right)$$

où ω_i^0 désigne le volume initial,

- lorsque ΔI_{max} est faible, on fait le bilan volumique pour tenir compte des déplacements sans remettre à jour les caractéristiques des interfaces,
- lorsque ΔI_{max} est grand (ou bien lorsque la somme des ΔI_{max} calculés depuis la dernière remise à jour des intersections est grande), on recalcule toutes les intersections (ainsi, le bilan volumique est exactement vérifié sans utiliser $\Delta\omega_i$).

Cette technique permet de diviser le coût des calculs d'intersections par un facteur très important (de l'ordre de cent).

Finalement, il reste à traiter le problème des mailles fortement couvertes.

En effet, lorsque le recouvrement d'une maille par le "masque" conduit à des volumes très faibles, la condition de CFL (9d) devient trop pénalisante puisque le pas de temps doit tendre vers zéro. La solution consiste à assembler ces mailles avec des mailles voisines suffisamment découvertes. De cette façon, l'ensemble formé d'une maille suffisamment découverte et de ses associés constitue une "macro-maille" dont le volume est assez grand pour ne plus pénaliser le pas de temps.

Nous utilisons comme critère le taux de couverture: une maille doit être assemblée lorsque sont volume est couvert à plus de 70%. D'autre part elle sera assemblée avec le voisin qui possède avec elle en commun l'interface la plus grande et qui est découvert à plus de 30% en volume.

Lorsque l'assemblage n'est pas possible directement, une procédure itérative est mise en oeuvre: on assemblera alors par l'intermédiaire d'une cellule qui elle même est assemblée (...).

Lorsque l'assemblage n'est pas possible du tout, on évince du calcul les mailles incriminées.

APPLICATIONS

Nous présenterons ici des cas de calcul illustrant les possibilités de la méthode.

Les figures 3 et 4 montrent le type d'applications traitées grâce au code de calcul FLUSEPA. Il s'agit de simulations tridimensionnelles. Le Mach externe est compris entre 5 et 6.

Pour la séparation d'étage de missile sous incidence (figure n° 3), la période simulée est d'environ 150 ms et la durée du calcul est de 12 heures en pas de temps global. Notons que ce type de simulation nécessite aussi bien le calcul de l'écoulement externe que de l'écoulement inter étage puisque ils interagissent très fortement entre eux. D'autre part, dans l'inter étage, les pressions peuvent devenir importantes (lorsque la section de passage vers l'extérieur est faible). De ce fait, l'écoulement dans la tuyère peut être fortement décollé: il faut donc impérativement le calculer aussi.

Pour le largage d'accélérateurs (figure n° 4), la période simulée est d'environ 1,1 seconde. Les maillages comportent environ 100 000 mailles et la durée du calcul est d'environ 40 heures en temporel adaptatif (sur Cray YMP). Le gain de temps par rapport à un schéma à pas de temps global est d'environ un facteur 20. Afin de souligner la robustesse de la méthode, nous précisons que les phénomènes rencontrés lors de cette simulation sont fortement instationnaires (acoustique...) et que dans les zones de chocs forts, les rapports de pression sont de l'ordre de 4 000.

Notons pour finir que des études de validation (aussi bien bidimensionnelles que tridimensionnelles) comprenant des comparaisons avec des mesures expérimentales ont été menées avec succès.
Afin de réduire notre exposé, nous ne les présenterons pas ici.

CONCLUSIONS

Les démarches théoriques que nous avons menées nous prouvent aussi bien la consistance que la stabilité linéaire de la méthode sur des maillages multidimensionnels non réguliers en ,espace et en temps. L'expérimentation numérique nous a démontré le bon comportement des schémas lors de la résolution de systèmes non linéaires.
Quant à la précision de la méthode, elle est d'ordre 2 en espace et en temps sur les maillages d'hexaèdres structurés réguliers et d'ordre 1 au moins ailleurs.
Nous noterons finalement que le potentiel d'évolution du code est important puisque, par exemple, il est envisageable d'adapter le maillage par déformation (étant donnée notre formulation A-L-E.), par enrichissement (nous travaillons en non structuré) ou grâce au chevauchement d'un maillage localement adapté à l'écoulement...

RÉFÉRENCES

(1) S.K. GODOUNOV,A. ZABRODINE,
M. IVANOV, A.KRAÏKO, G. PROKOPOV
"Résolution numérique des problèmes multidimensionnels de la dynamique des gaz"
Éditions MIR - Moscou

(2) M. POLLET
"Comparison of transport schemes for Navier-Stokes equations, application to rocket propultion"
7th International Conference on Numerical Methods in Laminar and Turbulent flow, Stanford USA,1991.

(3) B. VAN LEER
"Towards the Ultimate conservative Difference SchemeV. A Second Ordre Sequel of Godunov's Method".
J. Comput. Phys. 32, 1979.

(4) F. GODFROY, P. JACQUEMIN and F. JOUVE
"Three dimensional simulation of unsteady and inviscid flows using a second order finite volume method. Application to flows inside solid propellant motors"
Computing methods in applied sciences and engineering EDITION Glowinski, INRIA

(5) W.L. KLEB and J.T. BATINA·
"Temporal Adaptive Euler/Navier-Stokes Algorithm Involving Unstructured Dynamic Meshes"
AIAA Journal, Vol 30, n°8, 1992

(6) S.L. HANDCOCK
"Finite difference equations for PISCES 2 DELK, a coupled Euler-Lagrange continuum mechanics computer program"
Physics Internatonal Technical Memo - TCAM 76-2, 1976

(7) P. BRENNER
"Three-Dimensional Aerodynamics with Moving Bodies Applied to Solid Propellant"
AIAA paper 91-2304, 1991.

(8) P. BRENNER
"Numerical Simulation of Three-Dimensional and Unsteady Aerodynamics About Bodies in Relative Motion Applied to TSTO Separation"
AIAA paper 93-5142, 1993.

figure 1: chevauchement de maillage

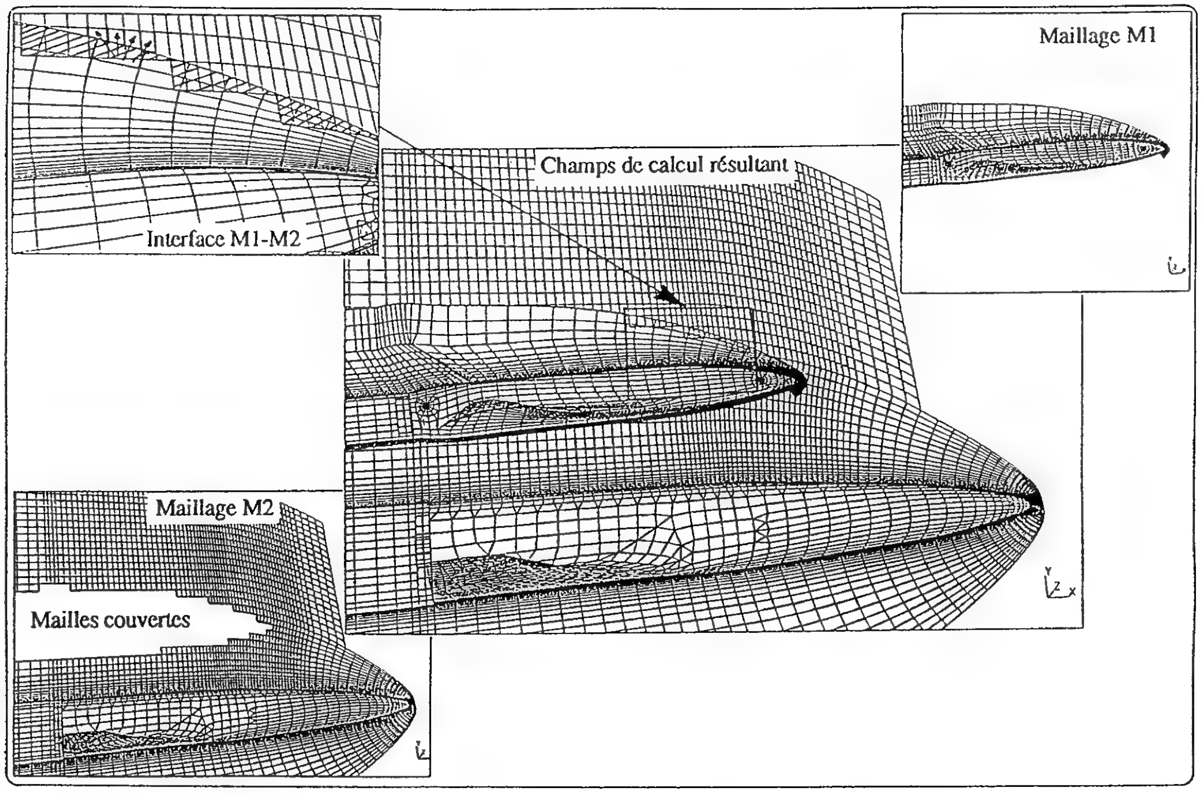


figure 2: priorité locale de recouvrement

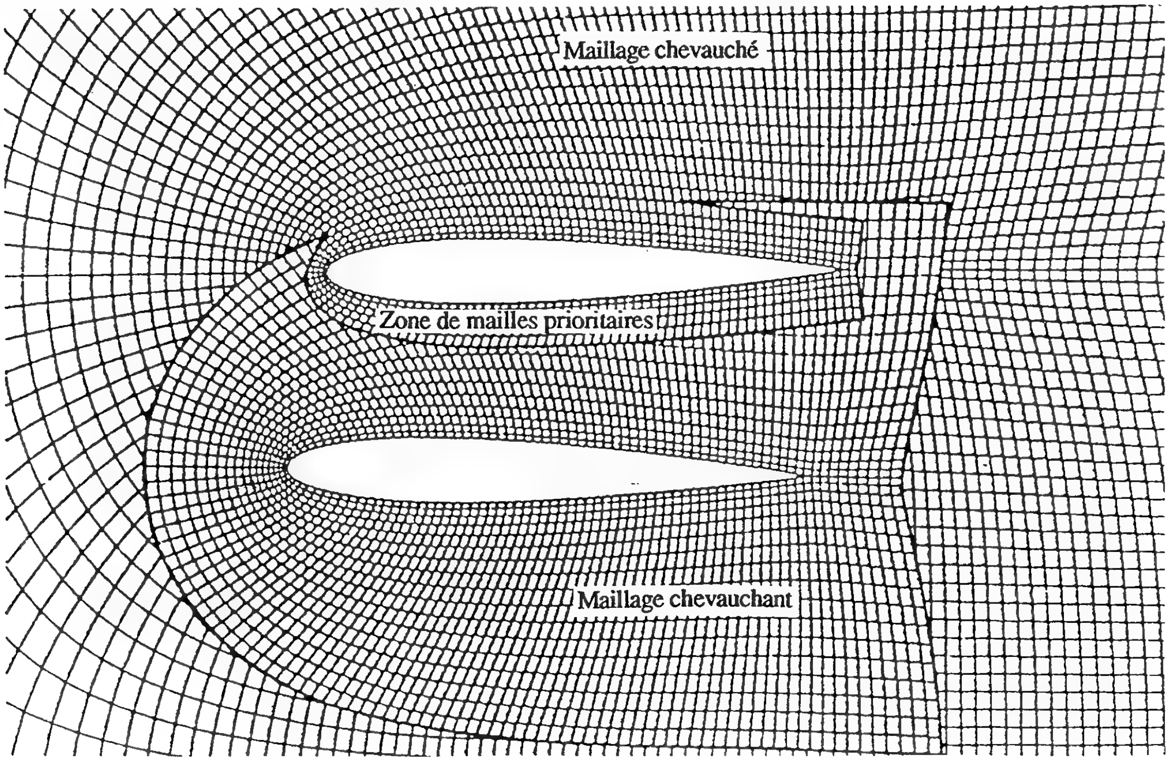


figure 3: séparation sous incidence par allumage direct

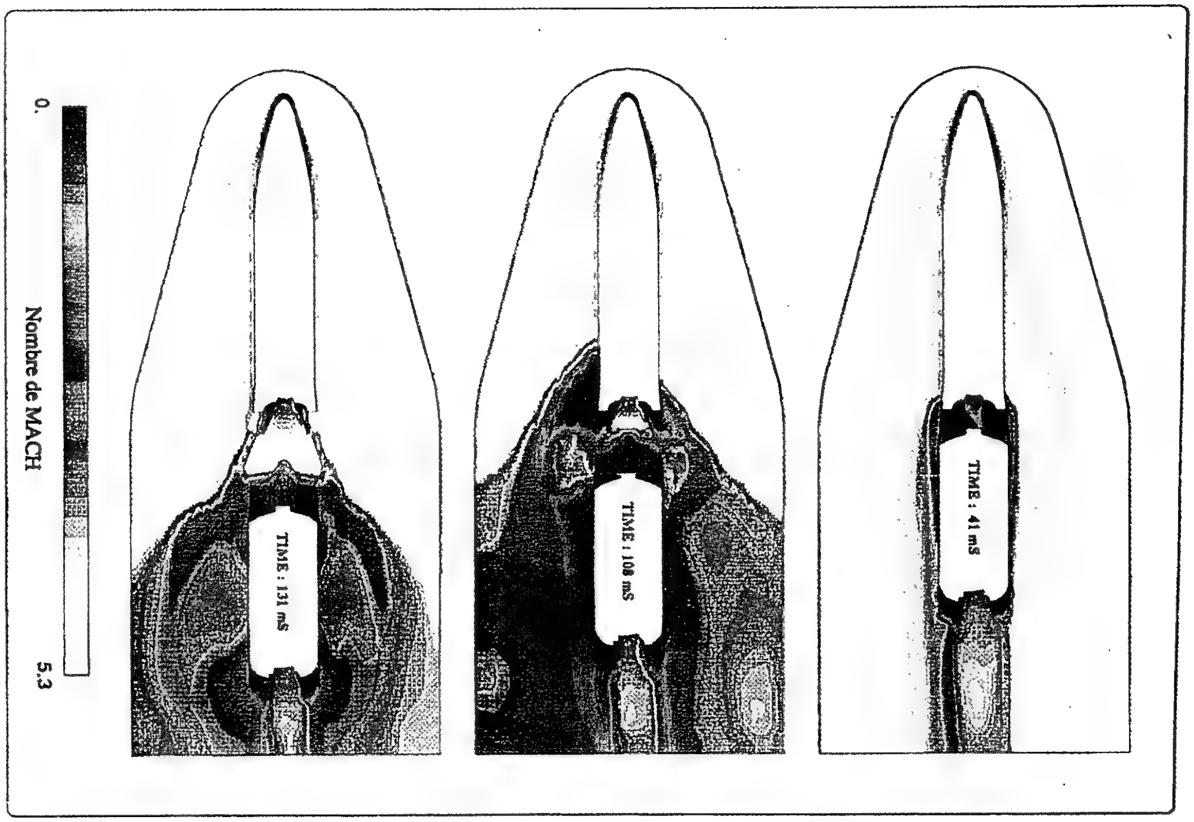
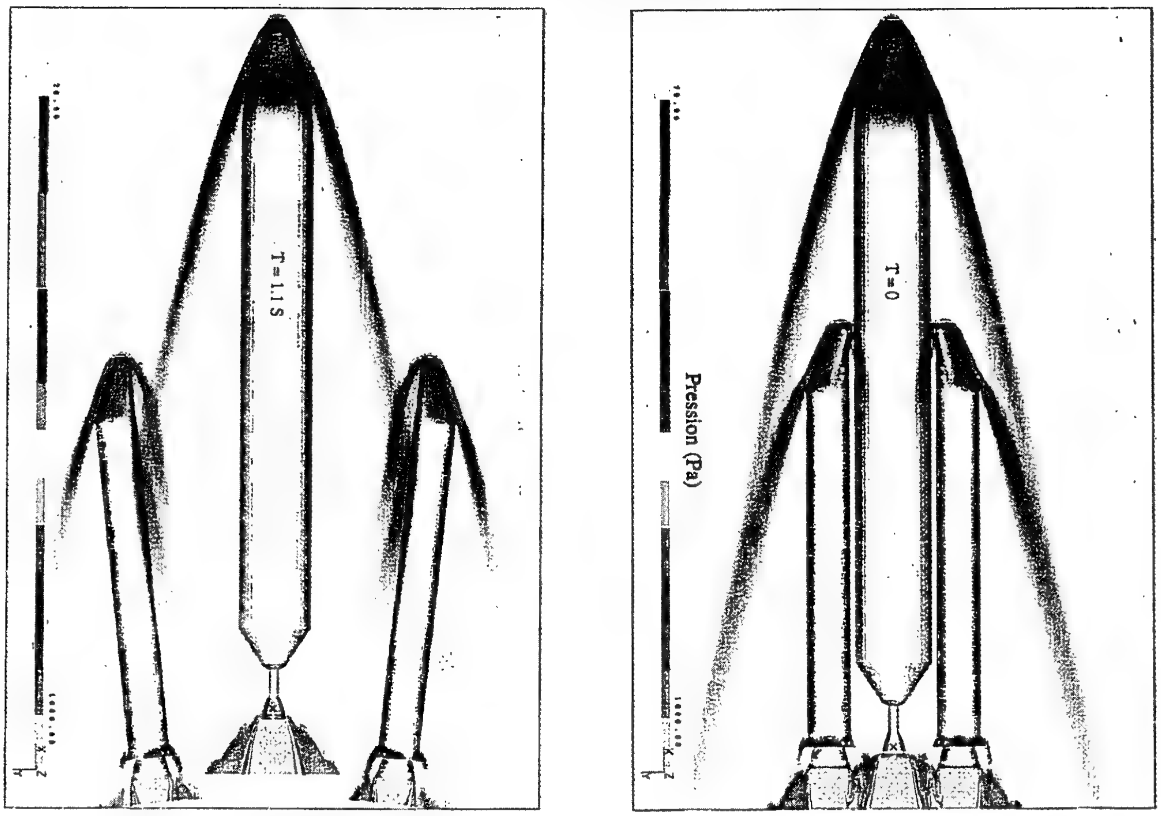


figure 4: largage d'accélérateurs par fusées d'éloignement



Efficient Numerical Simulation of Complex 3D Flows with Large Contrast

R. Radespiel, J.M.A. Longo, S. Brück*, D. Schwamborn*

DLR Institute of Design Aerodynamics
Braunschweig, Germany

*DLR Institute of Fluid Mechanics
Bunsenstrasse 10
37073 Göttingen, Germany

1. SUMMARY

Recent progress in flux vector splitting is reviewed with the aim to obtain high resolution and robustness for hypersonic reacting flow simulations. The numerical behavior of promising AUSM und CUSP discretization variants is reported and compared. These schemes can be combined with explicit multistage time stepping and multigrid. Large chemical source terms introduce stiffness into the system of equations which is removed by point implicit treatment. The present results demonstrate that efficient 3D simulations of viscous reacting flows with large contrast generated by strong shocks are now feasible.

2. INTRODUCTION

Accurate computations of 3D complex flow fields will play a key role in the aerothermal design of high speed vehicles such as reentry configurations. Not only can flow simulations shorten design cycles and save cost but they reduce uncertainty margins in heat loads and aerodynamic forces. A prominent example is the US-Orbiter vehicle, whose thermal protection system is heavy due to heat transfer uncertainties. Moreover, the space shuttle experienced an unexpected hypersonic pitch up which had not been predicted by conventional cold hypersonic wind tunnel testing.

The extensive use of 3D flow simulations for complex configurations within the aerodynamic design cycles has been precluded until recently by several reasons. Among these are long computation times of the codes simulating viscous flow or nonequilibrium chemistry. Also, many codes are not sufficiently robust in flow regions with strong shocks and flow expansions. Other codes are robust but they fail to resolve contact discontinuities such as boundary layers.

As a result of various attempts to solve complex hypersonic reacting flows numerically we can formulate some key requirements for the underlying solution algorithm. These are

- Capturing of strong shocks without oscillations of the dependent flow variables
- Robustness in regions of strong flow expansion
- Capturing of grid-aligned slip lines without numerical smearing
- Provision of an adaptive dissipative term in order to achieve sufficient numerical damping under adverse grid or flow conditions

The first requirement addresses the ability of the scheme to resolve complex 3D shock interactions with a limited amount of grid points. Moreover, oscillations at shocks may prevent convergence of the overall methods to the desired steady state solutions. The second point relates to the failure of various prominent discretization schemes when applied to rapid flow expansions into near vacuum conditions. Additionally, the scheme should resolve viscous shear layers with minimum numerical smearing in order to keep the number of grid points reasonably small. The fourth requirement results from the experience that one can usually not avoid adverse grid situations in 3D, particularly high values of cell aspect ratio. However, the available convergence acceleration techniques such as residual smoothing and multigrid rely on the damping of transient high frequency modes in the solution for which controlled artificial dissipation is necessary.

3. SHOCK CAPTURING, HIGH-ORDER SCHEME AND ADAPTIVE DISSIPATION

Progress in flux vector splitting has recently demonstrated that the aforementioned requirements can be fulfilled without characteristic decomposition of the inviscid flux and the corresponding matrix operations. The present paper covers two promising approaches into this direction which were initiated by Liou [1,2] and Jameson [3,4]. Other related pieces of work on the subject are found in Refs. [5-7]. The principal idea of flux vector splitting is shown by application to the 1D Euler equation

$$\frac{\partial W}{\partial t} + \frac{\partial F}{\partial x} = 0 \;, \quad W = \begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix} \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u H \end{bmatrix} \qquad (1)$$

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

Assume that the computational domain is discretized in intervals with the centers denoted by i-1, i, i+1, ... and the cell faces by i-1/2, i+1/2, ... Then, the numerical flux at the interface i+1/2 can be approximated according to Liou [1,2]

$$F_{i+\frac{1}{2}} = \frac{1}{2}M_{L/R}\left(c\begin{bmatrix}\rho\\ \rho u\\ \rho H\end{bmatrix}_R + c\begin{bmatrix}\rho\\ \rho u\\ \rho H\end{bmatrix}_L \right) - \frac{1}{2}|M_{L/R}|\left(\left(c\begin{bmatrix}\rho\\ \rho u\\ \rho H\end{bmatrix}_R \right) - \left(c\begin{bmatrix}\rho\\ \rho u\\ \rho H\end{bmatrix}_L \right) \right) + p_L^p + p_R^m \quad (2)$$

where L and R denote the right and left states at the cell face, $M_{L/R}$ is the upwind weighted Mach number at the cell face and p^p, p^m are Mach number weighted contributions of the left and the right pressure values. Upwind weighting is accomplished by proper polynomials of the local Mach number [2], by which the scheme is made purely upwind for supersonic flow whereas central differencing is obtained for $M \to 0$. This scheme is called AUSM.[1]

The alternate approach followed by Jameson [3] is to take a central average and subtract a diffusive term

$$F_{i+\frac{1}{2}} = \frac{1}{2}\left(F_i + F_{i+\frac{1}{2}}\right) - d_{1+\frac{1}{2}}$$
$$d_{1+\frac{1}{2}} = \frac{1}{2}\alpha c\,(W_R - W_L) + \frac{1}{2}\beta\,(F_R - F_L) \quad (3)$$

with $W = (\rho, \rho u, \rho H)^T$. Now, the diffusion coefficients α and β must be chosen such that upwinding is obtained in the supersonic range and $d \to 0$ for $M \to 0$. This scheme is called HCUSP.

Both approaches use scalar dissipation functions so that the computational expense of the overall method is proportional to N where N is the number of flow equations to be computed. Note, that there is particular motivation to use the equation (3) rather than (2) if the discretization is combined with explicit multistage time stepping. Then, very effective hybrid multistage schemes are at hand [8] for which the dissipation terms are only evaluated at m out of totally n stages.

The conceptual differences between both flux vector split schemes show up for the problem of resolving a stationary shock wave. Fig. 1 sketches the situation encountered in the analysis of AUSM. It is assumed that the states (L) and (R) fulfill the jump conditions. Equilibrium of the state (L) is obtained if the flux in between (L) and (R) is obtained by full upwinding. Also, the state (R) is in equilibrium if an upwind flux is used for the interface. This requirements can be fulfilled by defining the speed of sound at the shock

$$c_L = c_R = u_R \text{ or } c_L = c_R = \frac{c^*{}_L}{u_L} \quad (4)$$

for upwinding the flux where c* is the critical speed of sound. Hence, the state (R) is made supersonic and it is fully cancelled by the flux formulation (2). Generalization of equ. (4) for arbitrary flow direction and speed is given in Ref. [2]. This scheme is called AUSM$^+$.

Another way to shock resolution with AUSM was obtained by the observation that the highly dissipative flux of Van Leer [9] differs from AUSM by a dissipative term.

$$F_{\text{Van Leer}} = F_{\text{AUSM}} - \frac{1}{4}(M_R - 1)^2\left(c\begin{bmatrix}\rho\\ \rho u\\ \rho H\end{bmatrix}_R - c\begin{bmatrix}\rho\\ \rho u\\ \rho H\end{bmatrix}_L \right)$$

for $0 < M < 1$

(5)

in subsonic flow and it is identical in the supersonic region [5]. Smoothly captured shocks may then be obtained by defining

$$F_{\text{Hybrid}} = (1 - \omega)\,F_{\text{Van Leer}} + \omega F_{\text{AUSM}}$$

and ω depends on the second difference of the pressure in order to detect shocks. This scheme is called hybrid AUSM.

With HCUSP, on the other hand, the shock structure is analyzed according to Refs. [3,4] for a shock with a single interior zone shown in Fig. 2. Again, the states (L) and (R) satisfy the jump conditions and (L) is supersonic. Equilibrium of the shock is obtained if fluxes

$$f_{L/A} = f_L \text{ and } f_{A/R} = f_R$$

Then the flux balance for points L, A, R is zero. The condition at the entrance to the shock is fulfilled if the flow is supersonic at (L/A). The condition at the shock exit leads to a Hugoniot equation for a moving shock

$$f_R - f_A + \frac{\alpha c}{1+\beta}(W_R - W_A) = 0$$

This equation can be solved by Roe linearization [10] and yields the relation

$$\alpha c = (1+\beta)\,(c-u) \quad \text{for } 0 < M < 1. \quad (6)$$

between the dissipation coefficients.

Jameson [3] has used equ. (6) in order to derive a suitable form of dissipating coefficients α and β, i.e.

[1] The extension to multidimensions on structured grids is standard and may be found in Ref. [5]

$$\alpha c = |M| - \beta u$$

$$\beta = \max\left(0, \frac{u+\lambda^-}{u-\lambda^-}\right) \quad , \quad 0 < M < 1$$

by which central differencing is again approached for M→0. We note that equ. (6) is not respected for M<0.5 so that one can expect problems with shock capturing if the Mach number at interface (A/R) is less than 0.5.

The spatial accuracy of the flux vector split schemes depends on the determination of the left and right states at the cell interfaces. For a first-order scheme the flow quantities at the left and right states are given by their values at the neighboring mesh points, i.e. i, and i+i, respectively. In the present work higher order accuracy is obtained with the MUSCL approach by which the flow quantities are extrapolated to yield left and right states. The extrapolation function is designed such that the accuracy is limited to first order at discontinuities in order to guarantee shock capturing without spurious oscillations. Unfortunately, we find that the two flux vector split approaches described above should not be combined with the same extrapolation functions.

The AUSM scheme works well with the van Albada limiter function

$$u_L = u_i + \frac{1}{2} \frac{\left(\Delta_+^2 + \varepsilon\right)\Delta_- + \left(\Delta_-^2 + \varepsilon\right)\Delta_+}{\Delta_+^2 + \Delta_-^2 + 2\varepsilon} \qquad (7)$$

$$\text{where} \quad \Delta_+ = u_{i+1} - u_i \quad , \quad \Delta_- = u_i - u_{i-1}$$

We extrapolate the primitive variables and the total enthalpy using equ. (7). Extrapolation of the latter quantity is needed in the energy flux in order to allow steady state solutions with constant energy. Also, the parameter ε is made large if the contravariant velocity is smaller than a certain fraction of the speed of sound. Doing this, clipping within boundary layers and false interpolation values of the contravariant velocity components are avoided. Typical results of limiter applications for high Reynolds number viscous flows are shown in Fig. 3. The pressure contours at the rear part of RAE 2822 airfoil at transonic flow conditions shows oscillations near the edge of the turbulent boundary layer if limiting of the cartesian velocity components is applied in the traditional manner. These oscillations disappear if the limiting operator is switched off for small values of the Mach number in the contravariant coordinate direction. More technical details of the limiter can be found in Ref. [5].

Unfortunately, we find that the application of the van Albada limiter with the HCUSP scheme yields some preshock oscillations. Hence, we use the limiting function described in [4] for the HCUSP scheme. That function has also been extended to avoid limiting in low Mach number regions, by which the accuracy of the results is generally improved.

Finally, it is necessary to add controlled artificial dissipation in flow regions where the damping characteristics of the basic scheme are too bad in order to allow proper convergence to the steady state solutions. Fig. 4 shows the situation of a cell with high aspect ratio in two dimensions. In this case transient modes in the direction of the short cell edge will be well damped by an explicit time integration method whereas modes along the long side of the cell remain almost undamped. This problem can be solved by a modification of the advection function (AUSM scheme).

$$F^c = \frac{1}{2}|S|\left(M_{L/R}\begin{bmatrix}\text{sum}\\ \text{advection}\end{bmatrix} - \Phi\begin{bmatrix}\text{diff}\\ \text{advection}\end{bmatrix}\right)$$

where Φ is a function [5] of the spectral radii, λ, in the coordinate directions i and j so that

$$\Phi = |M_{L/R}| \quad \text{for } \lambda_i >> \lambda_j$$
$$\Phi = \delta \qquad\quad \text{for } \lambda_i << \lambda_j$$

Typical values of δ used in the present work are δ=1/4. This adaptive dissipation formulation makes sure that boundary layers are not numerically smeared but there is sufficient damping of modes in the direction of long cell sides. A similar formulation has been implemented into the HCUSP scheme.

The capabilities of the present discretization schemes for perfect gas flows with shocks and shear layers are assessed by computations of transonic and hypersonic two-dimensional flows. Fig. 5 compares distributions of pressure coefficient, total pressure loss and grid convergence of the aerodynamic coefficients for transonic inviscid flow over NACA 0012 airfoil. AUSM$^+$ and HCUSP yield comparable shock resolution whereas the hybrid AUSM appears to be more dissipative at the shock. The HCUSP scheme generates more entropy at the leading edge and lift and drag values converge somewhat slower with grid density as compared to AUSM. On the other hand HCUSP is more rapid with respect to the residual convergence as compared to AUSM. Typical convergence rates of the multigrid method described below are 0.90 for HCUSP and 0.94 for AUSM.

The resolution of very strong shocks and hypersonic shear layers is shown in Fig. 6. The Mach number contours obtained for inviscid flow around a blunted wedge demonstrate almost perfect shock capturing within one cell for AUSM$^+$ and HCUSP whereas hybrid AUSM needs one interior point for this case. The resolution of the thermal boundary layer which is displayed on the right part of Fig. 6 is similar for HCUSP and AUSM. We note that both schemes give much better results compared to a conventional central-difference scheme with a single scalar viscosity (not shown here).

4. OPERATOR SPLITTING AND IMPLICIT TREATMENT OF THE CHEMICAL SOURCE TERMS

For flows with nonequilibrium chemistry additional conservation equations with chemical source terms occur, which render the system of equation stiff if the time scale of the chemical reactions is significantly smaller than the fluid mechanics time scale. A simplified form of the conservation equation is given by

$$\frac{\partial}{\partial t}W = -F + S$$

$$W = (\rho, \rho u, \rho v, \rho w, \rho E, \rho_2, \ldots \rho_n)^T$$

$$S = (0, 0, 0, 0, 0, S_2, \ldots S_n)^T \quad , \text{ F=discr. flux}$$

The full set of equations used for reacting flow is given in Refs. [11, 12]. In order to overcome the time step limitations due to small chemical time scales we employ implicit discretization of the source terms,

$$\frac{\Delta W}{\Delta t} = -F^n + S^{n+1}$$

Using a linearization of the source term at time level (n) one obtains a point-implicit update of the solution vector W for the time level (n+1),

$$\frac{\Delta W}{\Delta t} = \left[I - \Delta t \frac{\partial S}{\partial W}\bigg|_n \right]^{-1} [-F^n + S^n] \qquad (8)$$

The Jacobian matrix has no entries in the first five rows because these equations have no source terms. Hence, the update of equ. (8) can be broken up into a fully explicit update for $W_1 = (\rho, \rho u, \rho v, \rho w, \rho E)^T$ followed by a point-implicit update for $W_2 = (\rho_2 \ldots \rho_n)^T$ which involves solution of (n-1) linear system for each grid point.

The evaluations of the explicit source vector, S^n, the elements of the flux jacobian, $\partial S / \partial W|_n$, and the solution of the linear system usually take much more computer time than the remaining elements of the solution algorithm. For multistage time stepping schemes the linearization of the source vector around the old time level is appropriate [13] and hence, the derivatives $\partial S / \partial W$ can be held constant during all stages.

5. MULTIGRID METHOD FOR HIGH SPEED FLOWS

Explicit multistage time-stepping schemes are used for advancing the solution in time. Choosing the number of stages and the stage coefficients allows an optimization of the high-frequency damping properties of the scheme at relatively high Courant numbers. Hence, these schemes can be combined with multigrid algorithms in order to accelerate convergence to steady-state, according to Ref. [8].

Coarse meshes for the multigrid are obtained eliminating alternate points in each coordinate direction. Both the solution and the residuals are restricted from fine to coarse meshes. A forcing function is constructed so that the solution on a coarse mesh is driven by residuals collected on the next finer mesh. The corrections obtained on the coarse mesh are interpolated back to the fine mesh. This multigrid scheme is now widely used in the CFD community and it works quite well for a wide range of subsonic and transonic flow problems.

However, a number of modest modifications of the original multigrid scheme are necessary for high Mach number flows with strong shocks and strong variations of viscosity and conductivity coefficients. We employ a special set of Runge-Kutta coefficients which are optimized for damping with upwind discretization and residual smoothing [14]. Courant numbers of about 5 are used in the present work which is about twice the explicit stability limit. Strong variations of viscosity and conductivity occur in hypersonic viscous flows. Typical time scales of the viscous diffusion process may be much smaller than the convection time scale which puts a severe restriction on the time step if purely explicit time integration is sought. This problem may be circumvented by locally adjusting the coefficient of the implicit residual smoothing scheme [15], such that the original time step based upon the inviscid flux vector is recovered,

$$\Delta t = \text{CFL} \frac{V}{|\lambda_\xi| + |\lambda_\eta| + |\lambda_\zeta|}$$

where λ_ξ denotes the spectral radius of the inviscid flux Jacobian in the ξ-coordinate direction and V is the cell volume. At strong shocks large Courant numbers obtained with the help of residual smoothing will result in solution divergence. Therefore an adaptive time step is employed such that the Courant number is reduced to about 1 at strong shocks [14].

The multigrid scheme involves restriction and prolongation operators which are both modified for hypersonic flows. At strong shocks the restriction of residuals from coarse to fine meshes is damped by using the second difference of the pressure as a sensor in order to reduce the coarse-mesh corrections in that region.

We have also observed that the coarse meshes can promote upstream propagation of transient modes if central interpolation is used for prolongation of the corrections. This problem is resolved by using an upwind biased interpolation of the corrections where the Mach number in the contravariant coordinate direction is used to define the bias [16].

6. NUMERICAL RESULTS FOR HYPERSONIC REACTING FLOWS

Comparisons between the different flux vector splitting variants have been presented for 2D calorically perfect gas flows within Chapter 2. Our experiences for reacting flows and for complex 3D flows are based upon the hybrid AUSM scheme, until now. The hybrid AUSM spatial discretization described in Chapter 2, the implicit source treatment given in Chapter 3, and the multigrid elements of Chapter 4 are implemented into the 3D DLR multiblock code CEVCATS [17, 5]. The implementation of the thermodynamic model, the chemical reactions and the viscous terms are general such that any chemistry model can be employed without modifications in the source code. Moreover, the code runs effectively on vector computer by vectorization over all grid points within a block of the computational domain. Inner loops containing the number of species or the number of reactions are unrolled by compiler directives [12]. Hence, we have obtained a computational speed of about 1500 MFLOP/s on a single processor of NEC-SX3 computer. This corresponds to 50 μs computing time for the update of a single grid point by one multigrid cycle and assuming a reacting gas mixture of five species with 17 chemical reactions. The use of point implicit operators and multigrid for reacting flows was also investigated with the help of a quasi 1D code for nozzle flows which contains the algorithmic elements presented in the previous chapters.

The capabilities of the multigrid method for reacting flows with large contrast are investigated by applications for one-, two- and three-dimensional flows. At first, we have chosen inviscid transonic reacting flow in a diverging nozzle in order to demonstrate the effects of point-implicit time stepping and multigrid acceleration separately. Fig. 7 displays the distributions of temperature and the concentrations of the three species present in the flow. The dissociation reaction rate coefficients for oxygen have been chosen such that strong reactions take place for temperatures above 1000 K. Hence, the flow simulation represents shock induced dissociation at reentry flow conditions. The dissociation time scale is small enough so that explicit time stepping alone does not yield a converged flow solution within several thousand time steps. With point-implicit time stepping the code converges slowly to the steady state. Convergence is noticeably accelerated by application of 4-level multigrid, by which a convergence rate per multigrid cycle of about 0.95 is realized.

The second application is the viscous reacting flow over a 2D cylinder which is displayed in Fig. 8. This case has been used to check the accuracy of reacting gas chemistry and thermodynamics into the 3D code CEVCATS by grid convergence studies and comparisons with other available codes (not shown here). It is found that both shock layer and wall heat fluxes are well predicted with relatively small numbers of grid points. The convergence histories plotted in Fig. 9 indicate again that multigrid is effective for reacting flow problems.

Finally, we present numerical results for a complex 3D case in Figs. 10-14. The configuration is called HALIS and it represents the windward side of the US-Orbiter vehicle. Extensive numerical and experimental data is available for this configuration. A first set of computations has been executed for wind tunnel conditions at Mach=10. These computations include the forebody, deflected body flaps and the base flow behind the rear of the vehicle. The computations were done on a mesh with 2.6 million grid points. Additionally, local grid refinement was investigated in the separation region around the deflected body flap. The numerical solutions have been extensively studied with respect to grid convergence and the solutions compare very well with wind tunnel measurements. The computation is a significant accomplishment because of the large regions with flow separation present.

The second computation was done for a flight trajectory point at Mach=24, 72 km altitude and assuming air in chemical nonequilibrium (see Figs. 12, 13). Until now, numerical results are obtained for the forebody of HALIS only. The grid convergence studies and the comparison with existing numerical data [18] indicates that the solution resolves the relevant flow phenomena properly. The residual convergence is displayed in Fig. 14 for both the nonreacting and the reacting flow cases. It is seen that the rate of convergence is approximately the same for both conditions. The computation of 400 multigrid cycles for HALIS forebody with 1.3 million grid points took 6 hours on NEC-SX3 computer. Hence, it is concluded that converged flow solutions for reacting flows over complex configurations are now feasible.

ACKNOWLEDGEMENT

The authors would like to thank R.C. Swanson for his cooperation on the 1D flow simulations. The communications with N. Kroll on the spatial and time discretization schemes are gratefully acknowledged.

7. REFERENCES

1. Liou, M.S. and Steffen, C.J.: *A New Flux Splitting Scheme*. J. Comput. Phys. 107, 23 (1993).
2. Liou, M.S.: *A Continuing Search for a Near-Perfect Numerical Flux Scheme, Part I: AUSM$^+$*. NASA TM 106524, 1994.
3. Jameson, A.: *Positive Schemes and Shock Modelling*

3. Jameson, A.: *Positive Schemes and Shock Modelling for Compressible Flows.* Lecture at 8th Finite Elements in Fluid Conference, Barcelona, 1993. To appear in Int. J. of Num. Methods in Eng.
4. Tatsumi, S.; Martinelli, L; Jameson, A.: *A New High Resolution Scheme for Compressible Viscous Flow with Shocks.* AIAA Paper 95-0466, 1995.
5. Radespiel, R. and Kroll, N.: *Accurate Flux Vector Splitting for Shocks and Shear Layers.* J. Comput. Phys., Vol. 121, No. 1, (1995).
6. Wada, Y. and Liou, M.S.: *A Flux Splitting Scheme with High Resolution and Robustness for Discontinuities.* AIAA Paper 94-0083, 1994.
7. Coquel, F. and Liou, M.S.: *Hybrid Upwind Splitting (HUS) by a Field-by-Field Decomposition.* NASA TM 106843, 1995.
8. .Jameson, A: *Multigrid Algorithm for Compressible Flow Calculations.* MAE Report 1743. Princeton University, October 1985. Text of Lecture given at 2nd European Conference on Multigrid Methods, Cologne.
9. Van Leer, B.: *Flux-Vector Splitting for the Euler Equations.* Lecture Notes in Physics 170, pp. 507-512, (1982).
10. Roe, P.L.: *Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes.* J.Comp. Phys. 43, pp. 357-372, (1981).
11. Brenner, G.: *Numerische Simulation von Wechselwirkungen zwischen Stößen und Grenzschichten in chemisch reagierenden Hyperschallströmungen.* DLR-FB 94-04, (1994).
12. Brück, S.; Radespiel, R.; Schwamborn, D.: *Extension of the Euler-/Navier-Stokes Code CEVCATS to Inviscid Nonequilibrium Flows.* DLR-Internal Report 223-95 A03, (1995).
13. Kroll, N. Private communication, (1994).
14. Radespiel, R. and Swanson, R.C.: *Progress with Multigrid Schemes for Hypersonic Flow Problems.* J. Comput. Phys., Vol. 116, (1995).
15. Radespiel, R. and Kroll, N.: *Multigrid Schemes with Semicoarsening for Accurate Computations of Hypersonic Viscous Flows.* DGLR Report 90-6, (1991).
16. Blazek, J.: *Verfahren zur Beschleunigung der Lösung der Euler- und Navier-Stokes-Gleichungen bei stationären Über- und Hyperschallströmungen.* DLR-FB 94-35, (1994).
17. Atkins, H.: *A Multiple-Block Multigrid Method for the Solution of the Three-Dimensional Navier-Stokes Equations.* DLR-FB 90-45, (1990).
18. Weilmuenster, K.J.; Gnoffo, P.A.; Greene, F.A.: *Navier-Stokes Simulations of the Shuttle Orbiter Aerodynamic Characteristics with Emphasis on Pitch Trim and Body Flap.* AIAA Paper 93-2814, (1993).

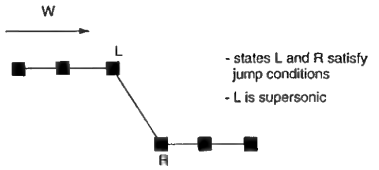


Fig. 1 AUSM$^+$ concept of exact shock capturing

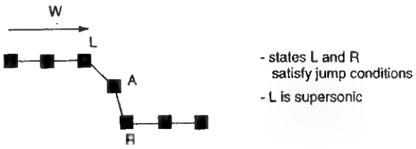


Fig. 2 CUSP shock concept of single interior point

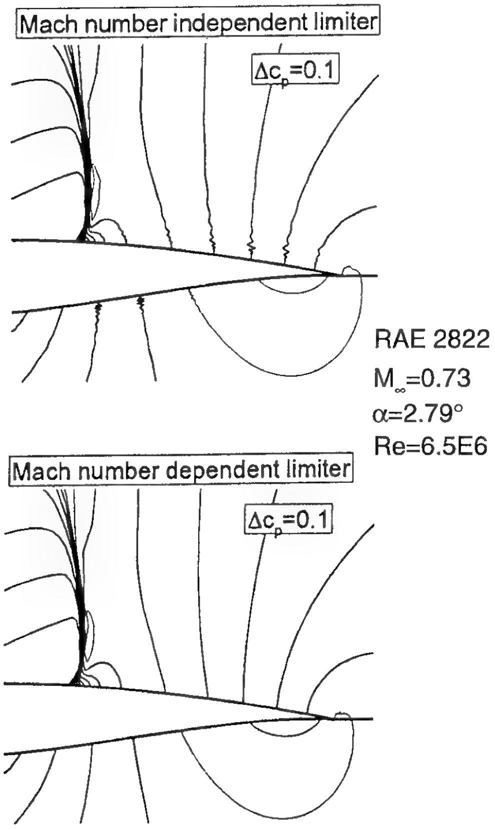
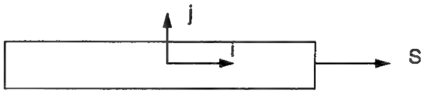


Fig. 3 Effect of false interpolation by traditional limiters for AUSM



the spectral radii of the inviscid flux Jacobians in i and j,

$\lambda_j >> \lambda_i$

Fig. 4 Cell with high aspect ratio in two dimensions

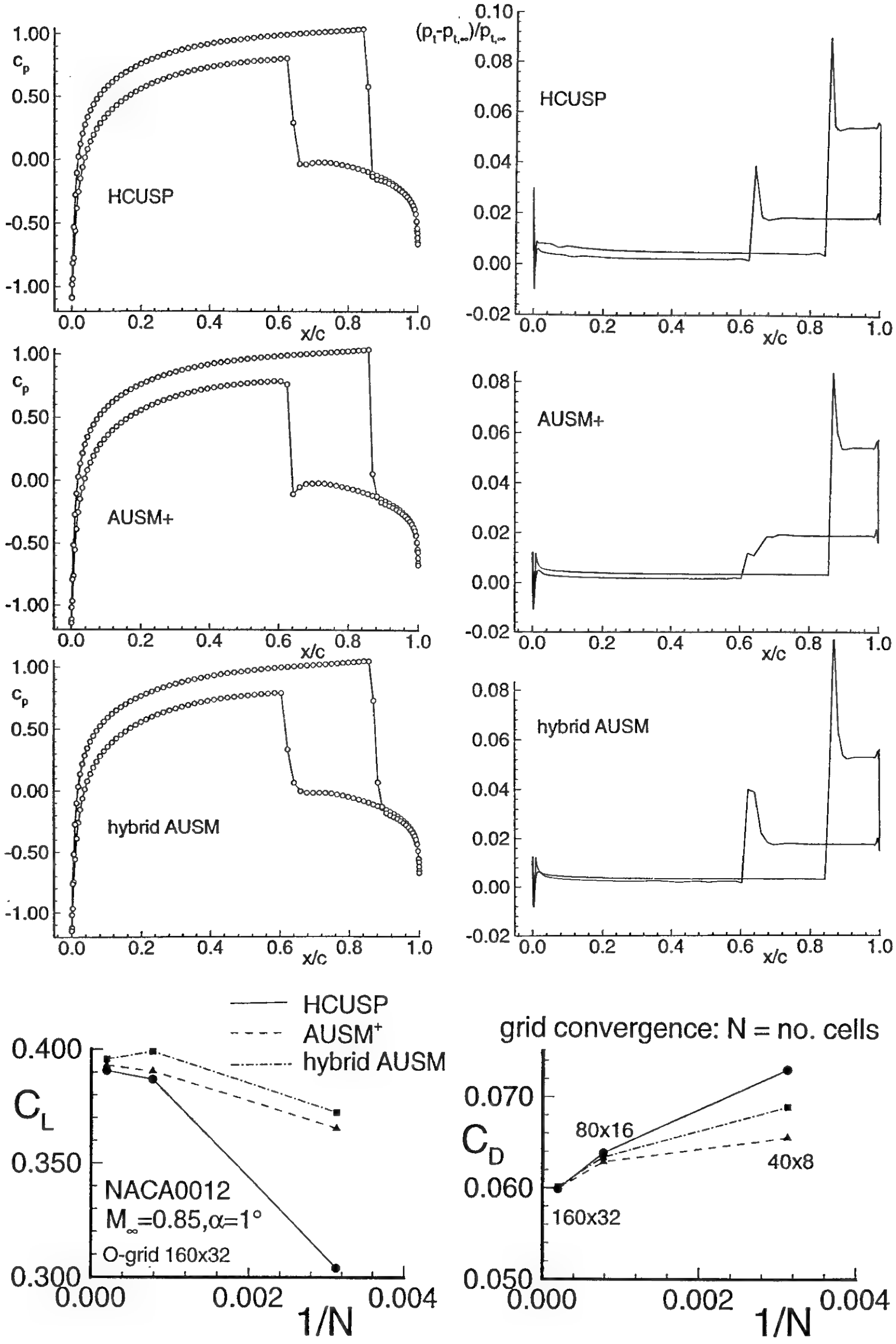
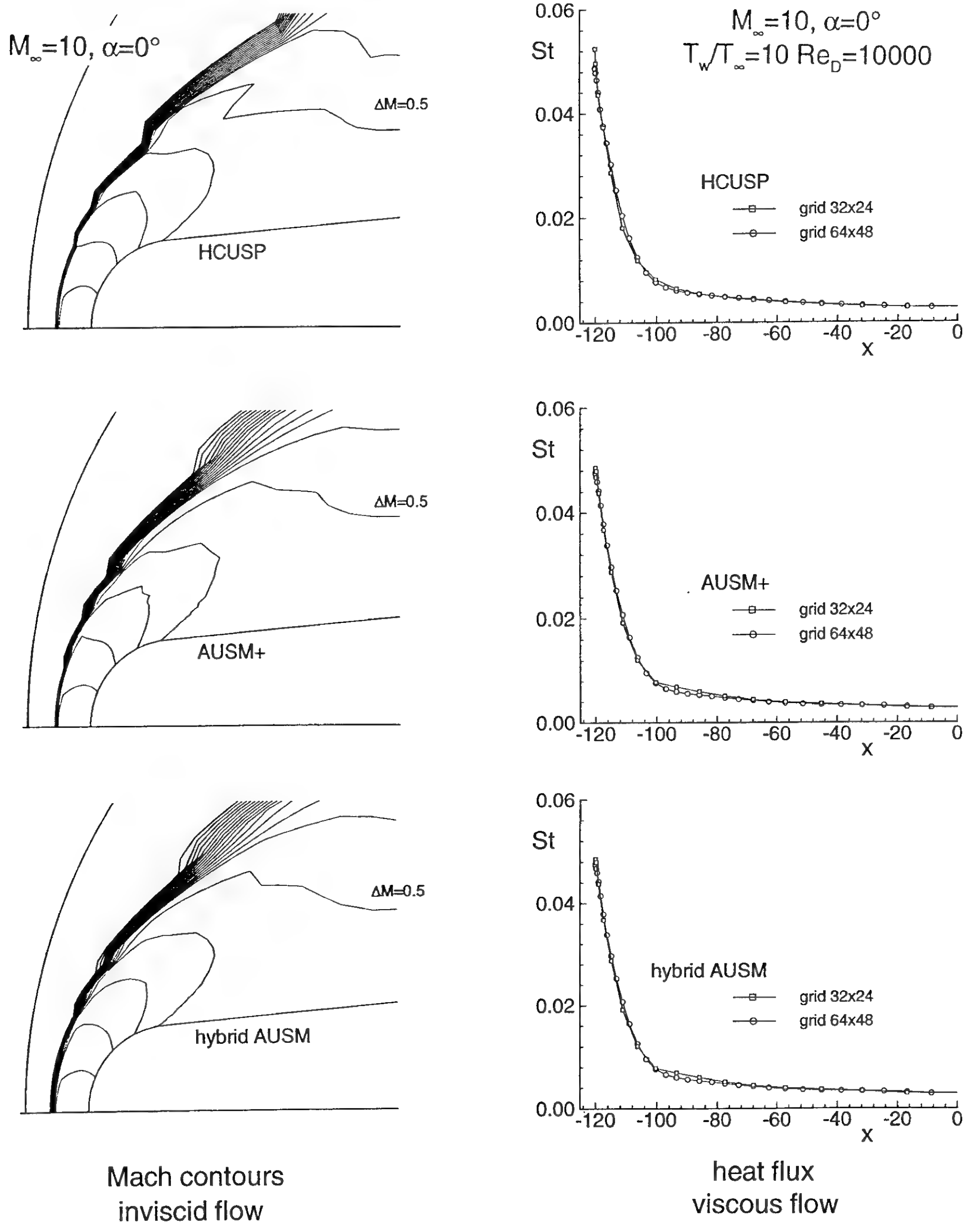


Fig. 5 Pressure, total pressure loss and grid convergence of total force for transonic inviscid airfoil flow



Mach contours
inviscid flow

heat flux
viscous flow

Fig. 6 Mach contours and wall heat flux for hypersonic flow over blunted wedge

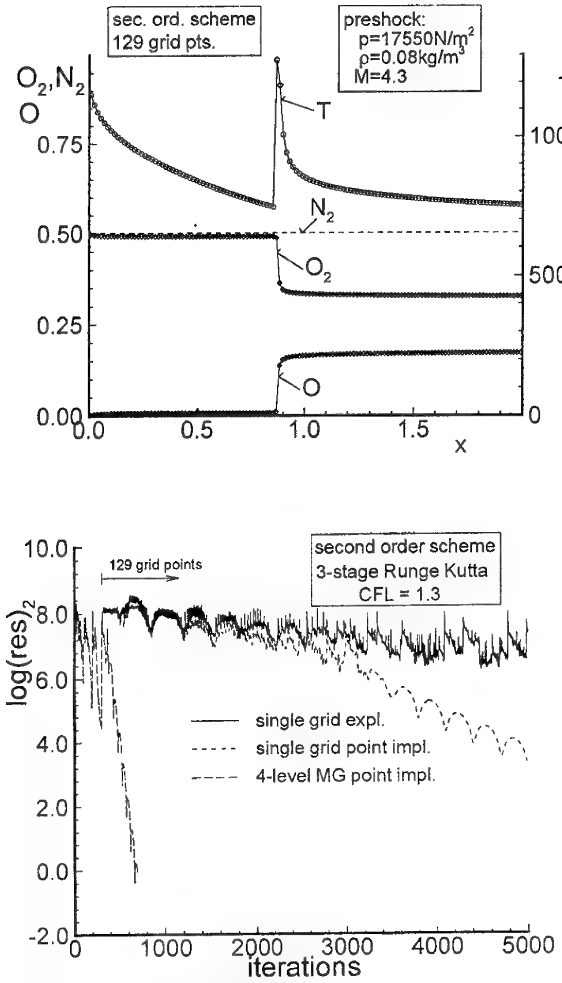


Fig. 7 Flow distribution and convergence histories of transonic reacting flow in diverging nozzle

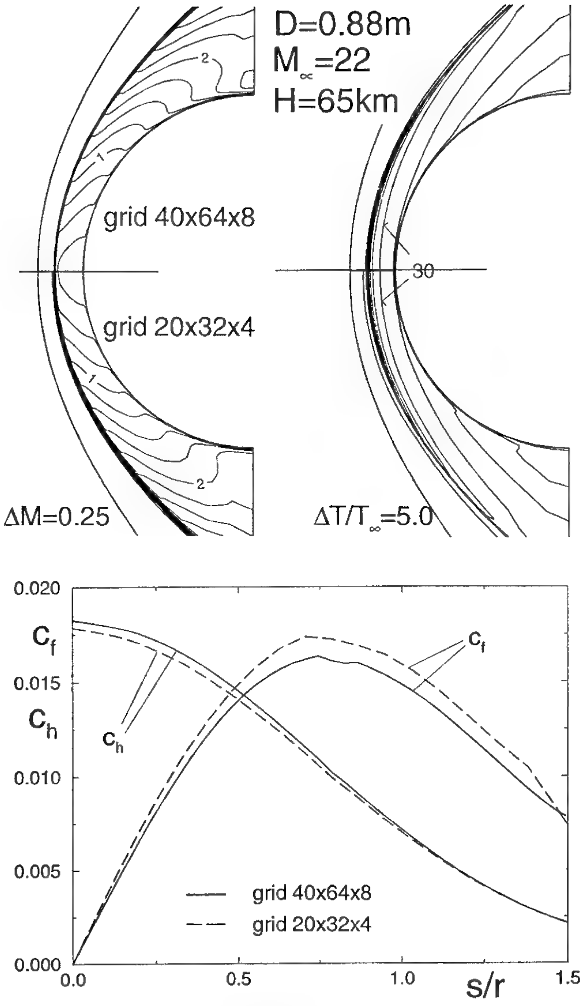


Fig. 8 Grid refinement study for flow of dissociating air over 2D cylinder

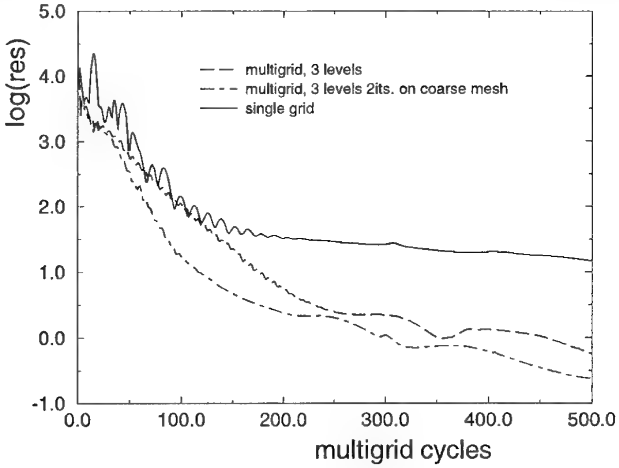


Fig. 9 Convergence histories for flow of dissociating air over 2D cylinder

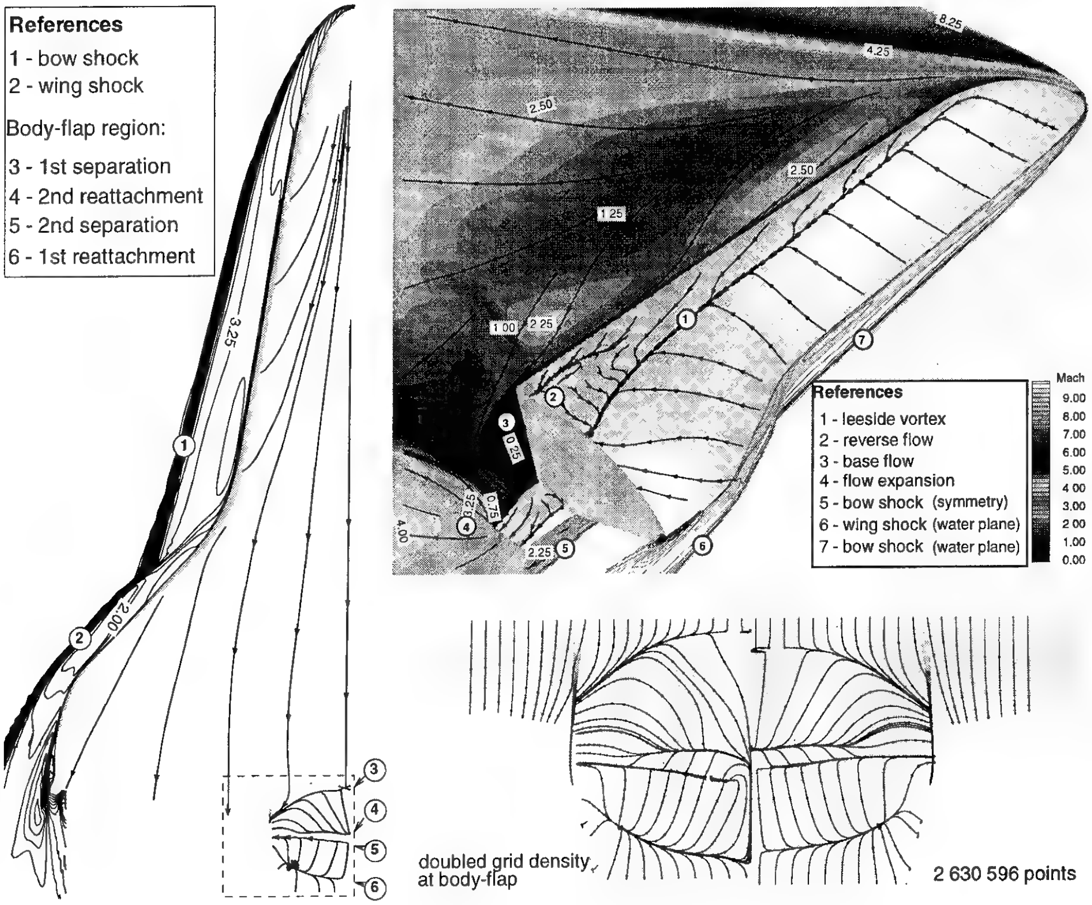


Fig. 10 Structure of 3D flow over HALIS reentry shape at wind tunnel conditions M_∞=10

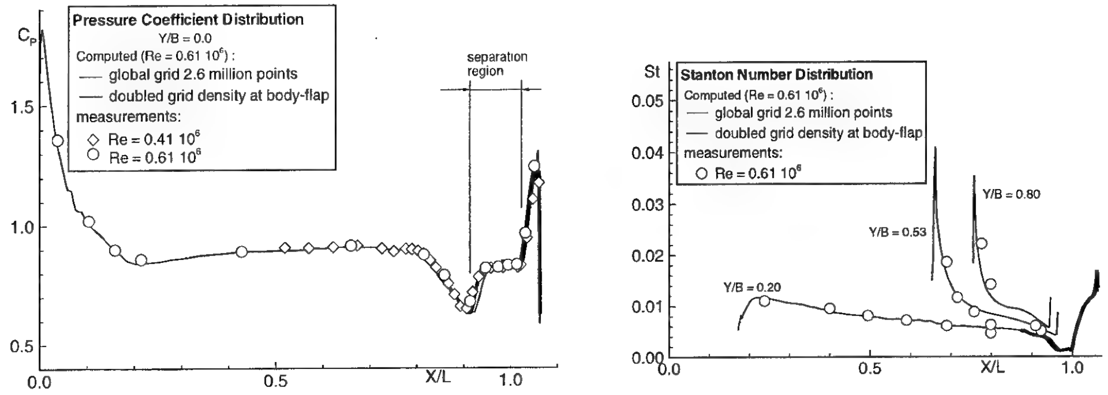


Fig. 11 Grid convergence study for HALIS and comparison with wind tunnel data in ONERA S4Ma

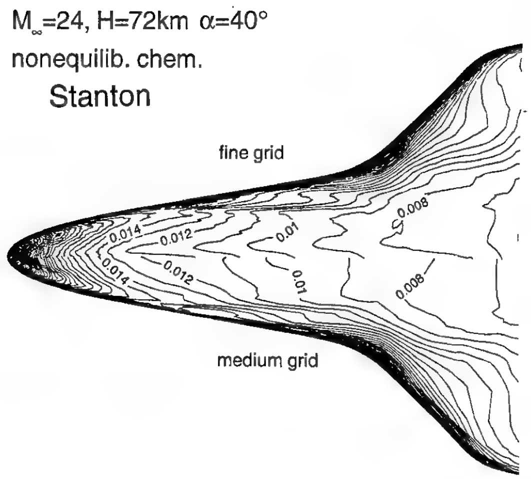


Fig. 12 Grid convergence for heat flux along windward side of HALIS forebody at flight trajectory point

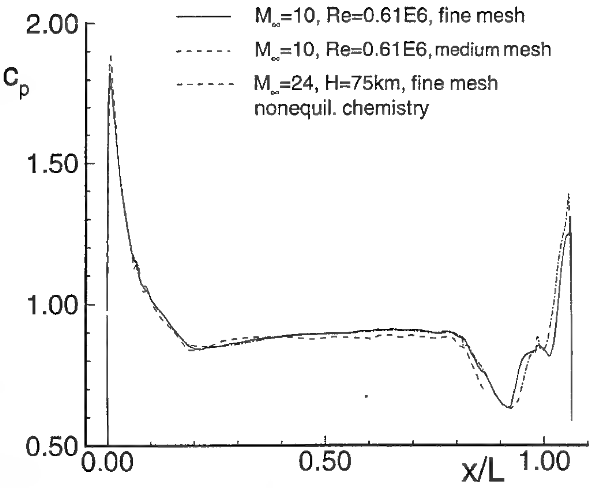


Fig. 13 Pressure distributions along windward symmetry line of HALIS

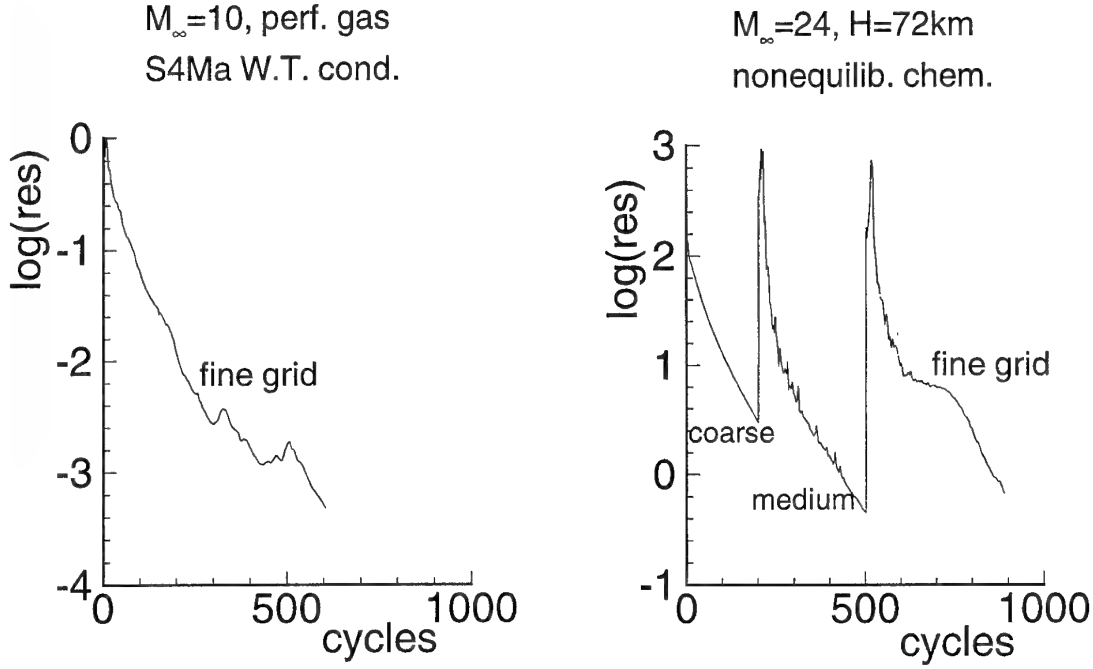


Fig. 14 Convergence histories for HALIS forebody computations

METHODES DE DECENTREMENT HYBRIDES POUR LA SIMULATION D'ECOULEMENTS EN DESEQUILIBRE THERMIQUE ET CHIMIQUE

Frédéric COQUEL [⋆], [†] , Véronique JOLY[†] et Claude MARMIGNON [†]
⋆ Laboratoire d'Analyse Numérique, CNRS URA 189, Université Paris VI, 75252 Paris Cedex 05
† Division de l'Aérodynamique théorique 1, ONERA, B.P 72 92322 Châtillon Cedex, France.

Abstract

The purpose of this paper is to present an hybrid upwind splitting method fully adapted to viscous chemical and thermal nonequilibrium flows. Such flows are the site of strong viscous-inviscid interactions and are dominated by real gas effects due to dissociation and internal mode excitation. Furthermore, the hypervelocities along the reentry trajectory induce a large degree of thermo-chemical nonequilibrium. ONERA has developed a code for simulating such flows: the code CELHYO. Detailed works concerning the physical modeling having already been presented in previous papers [4] [12], emphasis is put here on the numerical method, and particularly on the extension of hybrid upwind splitting methods to nonequilibrium flows. The hybrid upwinding is achieved by combining the basically distinct Flux Vector and Flux Difference Splitting approaches in retaining their own interesting features. The hybrid method implemented in the code CELHYO has been obtained by hybridizing the Osher approach with the van Leer scheme. In order to illustrate the numerical method, internal and external flow configurations are presented.

Résumé

On présente ici des méthodes numériques adaptées à la prédiction d'écoulements visqueux en déséquilibre thermodynamique et chimique. Cet article concerne en particulier le développement de schémas décentrés bien adaptés à l'approximation des flux de fluide parfait dans le contexte des problèmes d'écoulements visqueux hyperenthalpiques. Initialement, l'algorithme de traitement des flux de fluide parfait était dévolu à la méthode de Roe. Du fait de la mise en œuvre délicate de cette méthode dans le cadre des écoulements visqueux, une nouvelle approche pour le décentrement est proposée [8]. Elle combine les deux approches classiques, l'approche de décomposition de flux et l'approche de type Godunov. Nous soulignerons les principales étapes qui la composent et nous ne décrirons dans le présent papier qu'une technique d'hybridation particulière basée sur la méthode d'Osher et la méthode de van Leer. Afin d'illustrer la méthode numérique utilisée, des résultats de calculs pour des configurations d'écoulements externes et internes sont présentées.

1. Introduction

Nous nous intéressons à la résolution du système gouvernant les écoulements de gaz en déséquilibre thermique et chimique. De tels écoulements se produisent lors de la rentrée dans l'atmosphère d'un corps ou d'un véhicule hypersonique. A ces vitesses, l'écoulement atteint de très hautes températures près du véhicule. Ces températures sont suffisamment importantes pour induire des effets de gaz réels complexes comme la dissociation de l'air, la relaxation vibrationnelle et éventuellement l'ionisation. L' ONERA a développé un code de calcul simulant numériquement de tels écoulements: le code CELHYO. Des études détaillées relatives à la modélisation physique ayant déjà fait l'objet de plusieurs articles [4] [12], nous nous attachons ici aux travaux effectués dans le domaine numérique, et plus particulièrement à l'extension des schémas hybrides au cas d'écoulements en déséquilibre. La motivation de l'utilisation de tels schémas répond au souci de porter le code à un niveau de robustesse mais également de précision nécessaire à la simulation d'écoulements plus complexes correspondant par exemple aux écoulements ionisés.

Initialement, l'algorithme de traitement des flux de fluide parfait était dévolu à la méthode de Roe. Du fait de la mise en œuvre délicate de cette méthode dans le cadre des écoulements visqueux, une nouvelle approche pour le décentrement est proposée [8]. Elle combine les deux approches classiques, l'approche de décomposition de flux et l'approche de type Godunov. L'approche de décomposition de flux conduit à proposer des approximations simples se révélant être très

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

robustes dans la pratique mais présentant le principal défaut d'ignorer la structure de la solution relaxée, en particulier les ondes linéaires (discontinuités de contact) qui la composent. Le respect des ondes linéaires est crucial dans le cadre des écoulements visqueux et sa violation rend les méthodes de décomposition de flux inappropriées à leur simulation. L'approche de type Godunov permet de satisfaire cette exigence moyennant une complexité accrue de l'approximation. Mais elle trouve un avantage décisif sur l'approche de décomposition de flux dans le cadre des problèmes visqueux. Toutefois, ces méthodes peuvent présenter divers défauts de stabilité dans la capture des ondes non-linéaires (ondes de choc et de détente).

L'approche du décentrement par hybridation (méthodes HUS "Hybrid Upwind Splitting") combine les deux approches précédemment citées de manière à n'en retenir que les propriétés jugées idoines pour la simulation des écoulements visqueux hyperenthalpiques. Dans le code de calcul, c'est le schéma décentré résultant de l' hybridation du schéma d'Osher et de celui de van Leer qui a été implanté. D'autre part ont également été implantées dans le code la méthode d'Osher et une méthode de type van Leer.

2. Modélisation et équations de bilan

Dans cette étude, nous considérons un mélange idéal de gaz parfaits constitué de ns espèces dont nm espèces moléculaires. Dans le cas de l'air, les cinq espèces principales N_2, O_2, NO, N et O seront prises en compte. Les modes de translation et de rotation, et le mode électronique sont toujours considérés à l'équilibre et sont donc caractérisés par une température unique T alors que les modes de vibration peuvent s'écarter de l'équilibre. Nous supposons que parmi les nm espèces moléculaires, nv, $nv \leq nm$, d'entre elles ont leurs modes de vibration en déséquilibre (N2, O2 et éventuellement NO pour l'air). Nous nous intéressons aux évolutions bidimensionnelles de ce mélange. Ces évolutions sont gouvernées par le système de lois de conservation suivant:

$$\partial_t \mathbf{u} + \mathbf{div}(\mathbf{f}(\mathbf{u}) - \mathcal{D}(\mathbf{u})\mathbf{grad}\mathbf{u}) = \mathbf{\Omega}. \tag{1}$$

$\mathbf{f}$ désigne les flux de fluide parfait. Les phénomènes dissipatifs sont ici modélisés par le tenseur $\mathcal{D}$. Le terme source $\mathbf{\Omega}$ traduit la présence des phénomènes de déséquilibre. Dans la suite, U ouvert de $\mathcal{R}^p$ avec $p = ns + nv + 3$ désigne l'espace des états.L'inconnue $\mathbf{u} : \mathcal{R}^+ \times \mathcal{R}^2 \rightarrow \mathbf{U}$ a pour expression:

$$\mathbf{u}^T = \big(\, (\rho_\alpha)_{1 \leq \alpha \leq ns}, \; \rho v_1, \; \rho v_2, \; \rho E, \; (\rho_\beta e_{v;\beta})_{1 \leq \beta \leq nv} \big), \tag{2}$$

où E est l'énergie totale du mélange par unité de masse et $\mathbf{v} = (v_1, v_2)$ la vitesse barycentrique du mélange. $e_{v;\beta}$ désigne l' énergie de vibration par unité de masse de l'espèce moléculaire β.
La pression du gaz est donnée par la loi de Dalton:

$$p = \sum_\alpha \frac{\rho_\alpha}{M_\alpha} R_{gp} T, \tag{3}$$

où R_{gp} désigne la constante universelle des gaz parfaits et M_α est la masse atomique de l'espèce α.
A ce système est associée une relation de fermeture thermodynamique générale telle que la pression du mélange vérifie:

$$\begin{aligned} p = \kappa_{tr} \big(E - \frac{1}{2} \rho \mathbf{v}^2 - \sum_\beta \rho_\beta e_{v;\beta} \\ - \sum_\alpha \rho_\alpha (h_\alpha^0 + e_\alpha(T)) \big) \end{aligned} \tag{4}$$

où $\kappa_{tr} = \gamma_{tr} - 1$. e_α et h_α^o désignent respectivement l' énergie des modes internes à l'équilibre avec la température de translation et la chaleur de formation de l'espèce α par unité de masse.

Les expressions détaillées des termes source et du tenseur des phénomènes dissipatifs peuvent être trouvées dans de précédents articles [4], [12]. Nous rappellerons seulement que le modèle de chimie choisi est celui de Gardiner [9]. Il met en œuvre 17 réactions comprenant quinze réactions de dissociation et deux réactions d'échange. Les équations pour les énergies de vibration peuvent inclure les échanges d'énergie Translation-Vibration (T-V), Vibration-Vibration (V-V) ou Vibration-Dissociation (V-D). Le taux d'échange d'énergie T-V est modélisé par un modèle de Landau-Teller, les temps de relaxation entre espèces étant donnés par la loi semi-empirique de Millikan et White [14].
Le tenseur des contraintes visqueuses utilise pour la viscosité du mélange le modèle d'Armaly et Sutton [2], la viscosité de chaque espèce étant déterminée par la relation de Blottner [3]. La vitesse de diffusion des espèces vérifie une loi de Fick et un coefficient de diffusion binaire. Les flux de chaleur du mélange et de vibration sont supposés suivre des lois de Fourier. Le détail des expressions des coefficients de conductivité thermique est donné dans [4].

3. Méthode numérique

Les solutions du système convectif-dissipatif (1) sont approchées par une méthode de volumes finis implicite écrite pour des maillages curvilignes. Cette méthode est décrite ci-dessous, tout d'abord dans sa formulation semi-discrète en espace puis dans sa formulation implicite en temps. Seules seront discutées ici les méthodes numériques ayant trait à l'approximation du système Euler extrait. La discrétisation de l'opérateur du second ordre traduisant les phénomènes dissipatifs s'appuit sur une méthode centrée, déjà présentée à l'occasion d'une précédente contribution à l'AGARD [12]. Les méthodes associées au système Euler et implantées dans le code CELHYO, font appel à diverses techniques de décentrement. Elles relèvent respectivement du décentrement par résolution approchée du problème de Riemann, du décentrement par décomposition de flux et enfin d'une technique originale qualifiée de décentrement par hybridation champ par champ. Cette dernière technique résulte d'une recherche menée en collaboration entre l'ONERA et la NASA [8]. Nous décrivons ci-après trois des schémas décentrés utilisés dans le code: le solveur de Riemann approché d'Osher-Solomon, la décomposition de flux de van Leer et enfin le schéma HUS résultant de l'hybridation des deux précédentes méthodes. Le code CELHYO dispose également d'autres schémas, en particulier ceux de Roe, de Godunov et de Collela-Glaz, que nous ne rapporterons pas ici.

3.1 Méthodes de volumes finis bidimensionnels

L'approximation numérique de l'inconnue $\mathbf{u}$ du système est obtenue à l'aide d'une méthode de volumes finis dont la formulation continue en temps s'écrit, en omettant les termes source et les phénomènes dissipatifs, pour une cellule K de frontière ∂K:

$$(\mathbf{u}_K)_t \; + \; \frac{1}{|K|} \sum_{e \in \partial K_e} f(\mathbf{u}_K, \mathbf{u}_{K_e}; n_{K_e})|e| \; = \; 0, \quad (5)$$

où $\mathbf{u}_K$ (respectivement $\mathbf{u}_{K_e}$) désigne la valeur constante de la solution approchée sur la cellule K (respectivement K_e). Par définition, la cellule voisine K_e possède l'arête commune e; $\mathbf{n}_{K,e}$ est la normale unitaire à e extérieure à l'élément K. L'application $f \; : \; U \times U \times \mathcal{R}^2 \to \mathcal{R}^p$ désigne un flux numérique bidimensionnel astreint aux conditions de conservativité et de consistance usuelles. En vertu de l'invariance par rotation des équations d'Euler, l'évaluation des flux numériques bidimensionnels est déduite de l'évaluation d'un flux numérique consistant avec un problème de Riemann monodimensionnel où le flux exact s'écrit:

$$\begin{aligned} \mathbf{f}_{\tau_1} = \quad & \big(\, (\rho_\alpha v_1)_{1 \leq \alpha \leq ns}, \; \rho {v_1}^2 + p, \; \rho v_1 v_2, \\ & \rho H v_1, (\rho_\beta e_{v;\beta} v_1)_{1 \leq \beta \leq nv} \big), \end{aligned} \quad (6)$$

avec τ_1 le premier vecteur de la base canonique de$\mathcal{R}^2$.

Considérons $f(\mathbf{u}_K, \mathbf{u}_{K_e}; \tau_1)$ un flux numérique consistant avec le flux exact $\mathbf{f}_{\tau_1}$. En introduisant la rotation envoyant le vecteur de base τ_1 sur la normale $\mathbf{n}_{K,e}$, nous définissons l'opérateur de rotation $T_{K,e}$ qui nous permet d'obtenir un flux numérique bidimensionnel en posant:

$$f(\mathbf{u}_K, \mathbf{u}_{K_e}; n_{K,e}) \; = \; {T_{K,e}}^{-1} f(T_{K,e} \mathbf{u}_K, T_{K,e} \mathbf{u}_{K_e}; \tau_1). \quad (7)$$

Dans la suite, nous remplacerons abusivement $\mathbf{f}_{\tau_1}$ par $\mathbf{f}$, ceci afin d'alléger les notations.

3.2 Principales propriétés du système

Sous des hypothèses thermodynamiques générales, le système est hyperbolique. La matrice jacobienne associée, notée $\nabla \mathbf{f}(\mathbf{u})$, est diagonalisable et possède p valeurs propres λ_k, $1 \leq k \leq p$ dont deux valeurs propres simples $\lambda_1 = v_1 - c$, $\lambda_p = v_1 + c$ et une valeur propre multiple $\lambda_k = v_1$, $2 \leq k \leq p-1$. Ici, $c = \sqrt{\gamma \frac{p}{\rho}}$ désigne la vitesse du son.

Dans le cas d'une thermodynamique générale, les invariants de Riemann associés aux 1 et p-champs vraiment nonlinéaires ne sont pas explicitement connus. Ils sont donnés par les équations différentielles suivantes:

$$d\,Y_\alpha \; = \; 0 \qquad 1 \leq \alpha \leq ns-1 \quad (8)$$

$$d\,e_{v;\beta} \; = \; 0 \qquad 1 \leq \alpha \leq nv \quad (9)$$

$$d\,\rho \; \mp \; \frac{d\,p}{c^2} \; = \; 0 \quad (10)$$

$$d\,v \; \pm \; \frac{d\,p}{c\rho} \; = \; 0 \quad (11)$$

Lorsque la relation de fermeture thermodynamique n'est pas une fonction linéaire de la température de translation, γ dépend de la température et les équations (10) et (11) ne peuvent être facilement intégrées. Dans ce travail, nous proposons d'intégrer de manière approchée les relations (10) et (11) en négligeant la dépendance en température de γ pour obtenir les invariants suivants:

$$(Y_\alpha)_{1 \leq \alpha \leq ns}, v_1 \pm \frac{2c}{(\gamma-1)}, \frac{p}{\rho^\gamma}, \; v_2, \; (Y_\beta e_{v;\beta})_{1 \leq \beta \leq nv}, \quad (12)$$

Concernant les k-champs linéairement dégénérés, les invariants sont v_1 et p.

3.3 Méthode d'Osher-Solomon

Cette méthode repose sur la résolution approchée du problème de Riemann obtenue en assimilant chaque onde simple à une onde de détente-compression. Elle est ainsi définie par la construction d'un chemin dans l'espace des états reliant $\mathbf{u}_L$ à $\mathbf{u}_R$ et obtenu en suivant dans le plan vitesse-pression les parties admissible et non admissible des courbes de détentes issues de ces deux états. L'ordre de parcours des courbes de détente retenu dans ce travail correspond à l'ordre naturel $(v-c, v, v+c)$. Un tel chemin, noté $\Phi(\mathbf{u}_L, \mathbf{u}_R)$ dans la suite, existe sous des conditions thermodynamiques générales tant qu'il n'y a pas cavitation. Il est composé de deux sous-chemins de type vraiment nonlinéaire (VNL) notés Φ_1 et Φ_3 et d'un sous chemin, Φ_2, de type linéairement dégénéré (LD) associé à la discontinuité de contact. Ce chemin une fois construit permet de définir complètement le flux numérique associé au schéma d'Osher-Solomon selon :

$$f^{OS}(\mathbf{u}_L, \mathbf{u}_R) = \frac{1}{2}\Big(\mathbf{f}(\mathbf{u}_L) + \mathbf{f}(\mathbf{u}_R) - \int_{\Phi(\mathbf{u}_L,\mathbf{u}_R)} |\nabla_u \mathbf{f}(\mathbf{u})| d\mathbf{u}\Big). \quad (13)$$

Nous renvoyons à [5] pour l'écriture détaillée de ce flux. Nous nous consacrons ici à l'exposé de l'algorithme de construction du chemin $\Phi(\mathbf{u}_L, \mathbf{u}_R)$ que nous avons associé au mélange de gaz qui nous intéresse. Nous renvoyons à Abgrall et Coll. [1] pour un autre procédé.

Désignons par $\mathbf{u}_L^\star$ et $\mathbf{u}_R^\star$ les états séparés par la discontinuité de contact se propageant à la vitesse $v^\star$. Ces états sont construits en résolvant le problème suivant, exprimant la conservation des invariants de Riemann et la continuité de la pression et de la vitesse à la traversée de la discontinuité de contact.

$$Y_\alpha|_L^\star = Y_\alpha|_L, \quad Y_\alpha|_R^\star = Y_\alpha|_R, \quad (14)$$

$$\frac{p^\star}{\rho_L^{\star\,\gamma_L}} = \frac{p_L}{\rho_L{}^{\gamma_L}}, \quad \frac{p^\star}{\rho_R^{\star\,\gamma_R}} = \frac{p_R}{\rho_R{}^{\gamma_R}}, \quad (15)$$

$$v^\star + \frac{2}{\gamma_L - 1} c_L^\star = v_L + \frac{2}{\gamma_L - 1} c_L, \quad (16)$$

$$v^\star - \frac{2}{\gamma_R - 1} c_R^\star = v_R - \frac{2}{\gamma_R - 1} c_R, \quad (17)$$

$$Y_\beta e_{v,\,\beta}|_L^\star = Y_\beta e_{v,\,\beta}|_L, \quad Y_\beta e_{v,\,\beta}|_R^\star = Y_\beta e_{v,\,\beta}|_R. \quad (18)$$

Il est aisé de voir que la résolution du précédent problème peut être ramenée à la recherche de $v^\star$, solution de l'équation

$$p_R^\star(v) - p_L^\star(v) = 0, \quad (19)$$

traduisant la continuité de la pression et de la vitesse à la discontinuité de contact. Ce problème une fois résolu conduit à la détermination des autres quantités. Ici, nous avons :

$$p_R^\star(v) = p_R\big(1 + \frac{\gamma_R - 1}{2}\frac{v - v_R}{c_R}\big)^{\frac{2\gamma_R}{\gamma_R - 1}}, \quad v \le v_R,$$

$$p_L^\star(v) = p_L\big(1 - \frac{\gamma_L - 1}{2}\frac{v - v_L}{c_L}\big)^{\frac{2\gamma_L}{\gamma_L - 1}}, \quad v \ge v_L.$$

L'application $p_R^\star(v) - p_L^\star(v)$ est strictement croissante et admet au plus une racine, notée $v^\star$. Afin de calculer cette racine, il est utile d'introduire $\tilde{v_L}$ et $\tilde{v_R}$ définies par

$$\tilde{v_L} = v_L + \frac{\gamma_L - 1}{2} c_L, \quad \tilde{v_R} = v_R - \frac{\gamma_R - 1}{2} c_R,$$

et de remarquer [5] que $v^\star$ s'exprime comme combinaison convexe de ces deux vitesses particulières. Il existe donc un réel $z^\star \in [0, 1]$ tel que

$$v^\star = z^\star \tilde{v_L} + (1 - z^\star)\tilde{v_R}. \quad (20)$$

Notons que $\tilde{v_L} - \tilde{v_R} > 0$ sauf précisément lorsqu'il y a cavitation. En utilisant (20), le problème de la recherche de la racine de l'équation (19) peut alors être reformulé en ces termes. Trouver le réel $z^\star \in [0, 1]$ solution de l'équation :

$$C_{L,R}(1 - z)^{\frac{2\gamma_R}{\gamma_R - 1}} - z^{\frac{2\gamma_L}{\gamma_L - 1}} = 0, \quad (21)$$

où nous avons posé

$$C_{L,R} = \frac{p_R}{p_L} \frac{\big(2c_R/(\gamma_R - 1)\big)^{\frac{2\gamma_R}{\gamma_R - 1}}}{\big(2c_L/(\gamma_L - 1)\big)^{\frac{2\gamma_L}{\gamma_L - 1}}} (\tilde{v_L} - \tilde{v_R})^{\frac{2\gamma_R}{\gamma_R - 1} - \frac{2\gamma_L}{\gamma_L - 1}}.$$

Lorsque $\gamma_L \neq \gamma_R$, l'équation précédente n'admet pas de racine explicite, sa détermination nécessite la mise en oeuvre d'une procédure itérative de type Newton. Afin d'en optimiser la vitesse de convergence, nous proposons de substituer à la résolution du problème (21) celle du problème équivalent suivant, présentant un très bon conditionnement. Trouver le réel $z^\star \in [0, 1]$ solution de l'équation $g(z) = 0$ où:

$$g(z) = \begin{cases} C_{L,R}^{\frac{\gamma_L - 1}{2\gamma_L}} (1 - z)^{\frac{\gamma_R(\gamma_L - 1)}{\gamma_L(\gamma_R - 1)}} - z, & \text{si } z < \frac{1}{2} \\ C_{L,R}^{\frac{\gamma_R - 1}{2\gamma_R}} (1 - z) - z^{\frac{\gamma_L(\gamma_R - 1)}{\gamma_R(\gamma_L - 1)}}, & \text{sinon} \end{cases}. \quad (22)$$

Afin d'initialiser l'algorithme de Newton, nous définissons

$$z_1 = \frac{C_{L,R}^{\frac{\gamma_R - 1}{2\gamma_R}}}{1 + C_{L,R}^{\frac{\gamma_R - 1}{2\gamma_R}}}, \quad z_2 = \frac{C_{L,R}^{\frac{\gamma_L - 1}{2\gamma_L}}}{1 + C_{L,R}^{\frac{\gamma_L - 1}{2\gamma_L}}}. \quad (23)$$

Il est possible de vérifier que la plus proche valeur de la racine $z^\star$ est donnée par

$$z_{init} = \begin{cases} \max(z_1, z_2), & \text{si } \frac{\gamma_L(\gamma_R - 1)}{\gamma_R(\gamma_L - 1)} < 1, \\ \min(z_1, z_2), & \text{sinon} \end{cases}, \quad (24)$$

valeur qui sera utilisée comme valeur d'initialisation. Remarquons que dans le cas $\gamma_L = \gamma_R$, $z_{init} = z_1 = z_2$

coïncide avec la racine z^* de l'équation considérée. L'algorithme (22)-(24) converge généralement en au plus 3 itérations pour un test d'arrêt de 10^{-6} portant sur l'erreur relative $|\frac{p_R^*}{p_L^*} - 1|$.

3.4 Méthode de décomposition de van Leer

L'extension de la méthode de van Leer aux équations d'Euler multi-espèces et multi-températures a fait l'objet de quelques travaux (voir en particulier [11]). Les extensions proposées conduisent à une famille de schémas à un degré de liberté paramètrant la décomposition de flux d'énergie. Le choix d'une décomposition de flux particulière doit être opéré de manière à assurer que les matrices jacobiennes des flux décomposés $\pm\nabla f^{\pm}(\mathbf{u})$ n'admettent que des valeurs propres réelles positives ou nulles. Toutefois, il ressort de ces travaux qu'une telle propriété est difficile à garantir pour tout $\mathbf{u}$ de l'espace des états dans le cadre d'une thermodynamique non linéaire en T. Nous avons privilégié dans le code CELHYO le représentant de la famille considérée permettant de préserver la constance de l'enthalpie totale à la traversée d'un choc stationnaire. Ce schéma, brièvement décrit ci-dessous, s'est révélé entièrement satisfaisant dans les applications pratiques.

En introduisant le nombre de Mach $M = v/c$, les flux décomposés se réduisent à $f^+(\mathbf{u}) = \mathbf{f}(\mathbf{u})$, $f^-(\mathbf{u}) = 0$ lorsque $M \geq 1$ et symétriquement à $f^+(\mathbf{u}) = 0$, $f^-(\mathbf{u}) = \mathbf{f}(\mathbf{u})$ lorsque $M \leq -1$. Pour $|M| \leq 1$, ces flux sont définis par les expressions suivantes :

$$f^{\pm}_{\rho_\alpha} = \pm\frac{1}{4}(M \pm 1)^2 \rho_\alpha c, \quad 1 \leq \alpha \leq ns, \tag{25}$$

$$f^{\pm}_{\rho v_1} = ((1 \pm \frac{1}{\gamma})M + \frac{2}{\gamma})c f^{\pm}_{\rho}, \tag{26}$$

$$f^{\pm}_{\rho v_2} = v_2 f^{\pm}_{\rho}, \tag{27}$$

$$f^{\pm}_{\rho E} = H f^{\pm}_{\rho}, \tag{28}$$

$$f^{\pm}_{\rho_\beta e_{v,\beta}} = \rho_\beta e_{v,\beta} f^{\pm}_{\rho}, \quad 1 \leq \beta \leq nv, \tag{29}$$

où nous avons posé

$$f^{\pm}_{\rho} = \sum_{1 \leq \alpha \leq ns} f^{\pm}_{\rho_\alpha}. \tag{30}$$

Bien que fort éloigné de la méthode décentrée d'Osher-Solomon, le schéma de van Leer peut néanmoins recevoir une formulation analogue en terme de chemin. Il est ainsi possible de vérifier que celui-ci peut s'écrire

$$f^{VL}(\mathbf{u}_L, \mathbf{u}_R) = \frac{1}{2}\Big(\mathbf{f}(\mathbf{u}_L) + \mathbf{f}(\mathbf{u}_R) - \int_{\Phi(\mathbf{u}_L,\mathbf{u}_R)} (\nabla f^+(\mathbf{u}) - \nabla f^-(\mathbf{u}))d\mathbf{u}\Big), \tag{31}$$

et ce pour n'importe quel chemin Φ connectant $\mathbf{u}_L$ et $\mathbf{u}_R$ dans l'espace des états. Cette propriété est à la base de la technique d'hybridation des méthodes d'Osher-Solomon et de van Leer [7] dont nous proposons l'extension ci-dessous au cadre des mélanges de gaz en déséquilibre chimique et thermique.

3.5 Méthode de décentrement par hybridation champ par champ

L'introduction du décentrement par hybridation a été motivée par l'analyse des avantages et des défauts respectifs aux schémas d'Osher-Solomon et de van Leer. Ainsi si la méthode de van Leer se révèle être très robuste dans la capture des ondes non linéaires (choc et détente), elle est en revanche très peu précise dans la résolution des ondes linéaires (discontinuité de contact). Ce manque de précision la rend inappropriée dans le contexte d'équations de fluides visqueux qui est le nôtre. A l'opposé, la méthode d'Osher-Solomon autorise par construction la résolution exacte des discontinuités de contact stationnaires. Toutefois associé à cet avantage, cette dernière souffre d'un manque de robustesse dans la capture d'ondes nonlinéaires intenses. Les avantages et les défauts inhérents aux deux méthodes se révèlent donc disjoints et complémentaires.

La technique d'hybridation se propose de tirer parti d'une telle complémentarité avec pour but d'associer la robustesse de la méthode de van Leer dans la résolution des ondes non linéaires et la précision du schéma d'Osher-Solomon dans la résolution des ondes linéaires. C'est ainsi que chacun des trois sous chemins composant le chemin d'Osher-Solomon est associé soit avec la méthode de van Leer soit avec la méthode d'Osher suivant la nature nonlinéaire du sous chemin considéré. Dans la suite, nous notons $VNL(\Phi) = \Phi_1 \cup \Phi_3$ et $LD(\Phi) = \Phi_2$. Le flux numérique résultant de l'opération d'hybridation trouve alors l'expression suivante

$$\begin{aligned} f^{HUS}(\mathbf{u}_L, \mathbf{u}_R) &= \frac{1}{2}\Big(\mathbf{f}(\mathbf{u}_L) + \mathbf{f}(\mathbf{u}_R) \\ &\quad - \int_{LD(\Phi)} |\nabla \mathbf{f}(\mathbf{u})| d\mathbf{u} \\ &\quad - \int_{VNL(\Phi(\mathbf{u}_L,\mathbf{u}_R))} (\nabla f^+(\mathbf{u}) - \nabla f^-(\mathbf{u}))d\mathbf{u}\Big). \end{aligned}$$

Notons que par construction, le flux hybride coïncide avec le flux d'Osher en présence d'une discontinuité de contact seule et inversement se réduit au flux de van Leer lorsque seules n'interviennent que des ondes nonlinéaires dans la décomposition en ondes approchée d'Osher-Solomon. En reprenant les notations du paragraphe 3.3, la relation précédente peut être explicitée en

$$f^{HUS}(\mathbf{u}_L, \mathbf{u}_R) = f^{VL}(\mathbf{u}_L, \mathbf{u}_R) + \begin{cases} -\big(f^-(\mathbf{u}_R^\star) - f^-(\mathbf{u}_L^\star)\big), \text{ si } v^\star > 0, \\ +\big(f^+(\mathbf{u}_R^\star) - f^+(\mathbf{u}_L^\star)\big). \text{ sinon} \end{cases} \tag{32}$$

Soulignons la réelle simplicité du flux hybride en comparaison à la méthode d'Osher originale. En particulier, les points soniques n'interviennent pas. De plus, l'unique test rentrant dans la formulation du flux hybride peut être automatiquement pris en compte en utilisant les propriétés de symétrie du flux de van Leer par rapport au nombre de Mach. Nous avons ainsi

$$f^{HUS}_{\rho_\alpha} = f^{VL}_{\rho_\alpha} + \big(f^+_{\rho_\alpha}(-|M_R^\star|) - f^+_{\rho_\alpha}(-|M_L^\star|)\big), \tag{33}$$

$$f^{HUS}_{\rho v_2} = f^{VL}_{\rho v_2} + \big(f^+_{\rho v_2}(-|M_R^\star|) - f^+_{\rho v_2}(-|M_L^\star|)\big), \tag{34}$$

$$f^{HUS}_{\rho E} = f^{VL}_{\rho E} + \big(f^+_{\rho E}(-|M_R^\star|) - f^+_{\rho E}(-|M_L^\star|)\big), \tag{35}$$

$$f^{HUS}_{e_{v,\beta}} = f^{VL}_{e_{v,\beta}} + \big(f^+_{e_{v,\beta}}(-|M_R^\star|) - f^+_{e_{v,\beta}}(-|M_L^\star|)\big), \tag{36}$$

et

$$f^{HUS}_{\rho v_1} = f^{VL}_{\rho v_1} - sign(v^\star)\big(f^+_{\rho v_1}(-|M_R^\star|) - f^+_{\rho v_1}(-|M_L^\star|)\big). \tag{37}$$

Notons qu'à l'instar du schéma d'Osher-Solomon, le flux hybride ne permet pas de satisfaire à un principe du maximum sur les fractions massiques et les énergies de vibration massique. La correction proposée par Larrouturou [10] peut lui être appliquée sans dégrader ni la robustesse ni la précision. Soulignons enfin qu'il est possible de donner à la technique d'hybridation un cadre beaucoup plus général que celui exposé ici [8].

3.6 Méthode du deuxième ordre explicite

La procédure d'extension de la méthode de volumes finis au second ordre d'approximation en espace est la méthode classique MUSCL de van Leer qui, à chaque pas de temps, repose sur une reconstruction affine par morceaux de la solution approchée. L'extension de la méthode MUSCL que nous utilisons au cas d'un mélange de gaz permet de respecter la conservation des espèces élémentaires et également d'assurer la positivité des fractions massiques et des énergies de vibration massiques sous certaines conditions de type CFL dans le cas d'un schéma explicite [13]. Dans le contexte des maillages curvilignes qui est le nôtre, la méthode est appliquée direction curviligne par direction curviligne. La méthode utilise les variables suivantes:

$$\big((Y_\alpha)_{1 \leq \alpha \leq n_s},\ \rho,\ v_1,\ v_2,\ p,\ (Y_\beta e_{v;\beta})_{1 \leq \beta \leq n_v} \big), \tag{38}$$

en inhibant la procédure de reconstruction sur les fractions massiques. Cette stratégie permet de garantir la conservation des espèces élémentaires qui autrement serait généralement perdue à cause des nonlinéarites inhérentes à la procédure de reconstruction. La fonction limitrice considérée est la fonction proposée par van Albada ou la fonction minmod.

4. Méthode implicite

La construction du schéma implicite est obtenue par une linéarisation des flux numériques et des termes source. L'implicitation des termes de flux de fluide parfait est basée sur la méthode de Flux Vector Splitting de van Leer, et ce indépendamment du flux explicite utilisé. La robustesse obtenue est a priori peu sensible à la nature du schéma explicite. Le terme source est traité de manière centrée. Les termes de dérivées croisées sont négligés dans l'étape de linéarisation des flux de fluide visqueux. Un bon traitement implicite des conditions aux limites conditionnant la qualité d'accélération de la convergence vers l'état stationnaire, une attention particulière y a été apportée. L'opérateur implicite ainsi obtenu est linéaire et est résolu par une méthode itérative. Une telle méthode présente l'avantage d'être bien moins sensible au choix du pas de temps que ne le sont les méthodes par factorisation approchées. Elles évitent également la décomposition parfois inadéquate de la matrice jacobienne des termes source. La méthode itérative mise en œuvre est basée sur une stratégie de minimisation des résidus telle que la méthode GMRES. La méthode itérative convergeant d'autant mieux que le système est bien conditionné, une factorisation ILU est utilisée.

5. Résultats numériques

Afin d'illustrer les capacités de la méthode pour calculer des écoulements en déséquilibre dans des configurations variées, des calculs d'écoulements externes autour d'une configuration d'hyperboloïde plus volet et d'écoulements internes dans une tuyère qui équipe la soufflerie hyperenthalpique F4 de l'ONERA ont été réalisés.

5.1 Ecoulement dans une tuyère de la soufflerie F4

La soufflerie ONERA F4 peut être équipée de quatre tuyères différentes correspondant chacune à des régimes d'écoulements différents. Les conditions des calculs que nous présentons ici correspondent au cas test numéro 1 du "Fourth European High Velocity Database Workshop" (qui s'est tenu le 24-25 Novembre 1994 à Noordwijk). La géométrie est celle correspondant à la tuyère $n^o 2$.Sa longueur totale est de 3.42 m et le rayon du col est de 0.005m. Les conditions dans la chambre correspondent à une enthalpie totale réduite $\frac{Hi}{R_{gp}T_0}$ de 260 et une pression de 430 bar. La température de la paroi est de 300 K. Elle est supposée totalement catalytique jusqu'à une distance de 0.5 m en aval du col, puis noncatalytique après.

Le domaine de calcul a été divisé en huit parties. Dans le premier domaine, l'écoulement dans le convergent et dans la région proche du col est calculé. Ces résultats servent ensuite pour déterminer la solution dans les zones suivantes de l'écoulement hypersonique. Chaque domaine comprend 89x85 points.

Un calcul laminaire et un calcul turbulent, respectivement notés (1) et (2) ont été réalisés. Pour le cas turbulent, le modèle de turbulence utilisé est le modèle algébrique de Baldwin-Lomax et le point de transition est situé à 0.5 m en aval du col. Les résultats présentés ont été obtenus après 6000 itérations dans le premier domaine. Pour les autres domaines, 600 à 200 itérations environ suivant le domaine considéré ont été nécessaires. Le nombre de Courant peut atteindre une valeur de 500 (pas de temps global) dans les derniers domaines du divergent. Dans tous les cas, les résidus maxima décroissent au moins de 10 ordres de grandeur.

La figure 1 montre l'ensemble du maillage utilisé pour la tuyère. Les résultats pour les calculs laminaire et turbulent sont présentés sur les figures 2 à 6. Le champ des nombres de Mach est visualisé sur la figure 2 dans le cas de l'écoulement laminaire et montre une onde venant perturber l'écoulement proche de l'axe de la tuyère. La naissance de cette onde correspond à un point d'inflexion de la géométrie. Sur la figure suivante sont portées les distributions de températures le long de l'axe dans le cas laminaire ou turbulent, aucun effet notable de la prise en compte de la turbulence sur ces distributions ne pouvant être observé. Les distributions transversales de nombres de Mach pour le cas laminaire et le cas turbulent sont montrées sur la figure 4 en sortie de tuyère. On observe une bonne uniformité du nombre de Mach dans le noyau de l'écoulement.

5.2 Calculs d'écoulements externes

Deux séries de calculs ont été réalisées autour d'une configuration d'hyperboloïde plus volet. Cette géométrie a été proposée pour le cas test numéro 4 du Workshop. La longueur totale de la maquette est de 0.1114 m et l'angle entre le volet et l'axe est de 43.6 degrés. Le premier calcul correspond aux conditions de l'écoulement dans la tuyère n^o 2 de la soufflerie F4, l'enthalpie totale réduite étant égale à 122 et la pression génératrice étant de 441 bar (soit les conditions du cas test numéro 4):
T_∞=187 K;
$T_{v,N2}$=4078 K; $T_{v,O2}$=2485 K;
ρ_∞=1.557 $10^{-3} Kg/m^3$; u_∞=3934 m/s; T_w=300 K;
C_{N2}=0.7254; C_{O2}=0.1354; C_{NO}=0.0895; C_N=10^{-20};
C_O=0.0497.
La paroi est supposée totalement catalytique.
Le deuxième calcul correspond à des conditions en vol sur une géométrie homothétique de la précédente dans un rapport 1.4:
T_∞=268 K;
ρ_∞=2.608 $10^{-3} Kg/m^3$; u_∞=5083 m/s; T_w=1000 K;
P_∞=201.5 Pa.
La paroi est également supposée totalement catalytique.

Le même maillage est utilisé pour les deux calculs qui tiennent compte de la différence d'échelle. Il contient au total 401x110 points. Trois sous-domaines ont été utilisés afin de diminuer les temps de calcul et la taille nécessaire de la mémoire. Les domaines se recouvrent sur quatre points. Ces domaines (nez, région intermédiaire et région du volet) comprennent respectivement 80x110, 123x110 et 206x110 points. Pour le nez et la région intermédiaire, on obtient une décroissance des résidus quadratiques explicites de 8 ordres de grandeur après 2000 itérations. Le nombre de Courant atteint 10 pour le nez et 70 pour la deuxième zone. Dans la région du volet, une décroissance de 5 ordres de grandeur des résidus est obtenue après 20000 itérations et un nombre de Courant de 10. Notons que les résidus n'atteignent pas de plateau et continuent de décroître lorsque les calculs sont poursuivis. Cette convergence lente est due à la présence d'une importante zone de recirculation dans la région de volet.

Les résultats sont présentés sur les figures 5 à 17. Les figures 5 à 9 montrent des courbes d'isovaleurs du nombre de Mach et de la pression pour les deux calculs. Dans la région du volet, la zone de séparation est bien définie pour les deux calculs (figures 5 à 7). La figure 6 montre un agrandissment de cette zone pour le cas du vol. Les effets visqueux sont importants du fait du faible rayon du nez. Des oscillations légères sur les courbes d'isopression sont visibles et correspondent à des sauts de mailles dans la région du choc. La pression atteint la valeur maximale de 22432 Pa pour le cas en soufflerie et 63073 Pa pour le cas du vol.

L'écoulement est relativement figé derrière le choc, comme le montrent les profils de température obtenus pour le premier calcul (figure 10). La distance du choc est dans ce cas égale à 3.7 10^{-4} m. Les figures suivantes montrent des valeurs à la paroi pour les deux calculs.

Le nombre de Stanton le long du corps est présenté sur la figure 11 pour le calcul en soufflerie. La valeur maximale (0.284) correspond à un flux de chaleur égal à 1.43 10^7 W/m2. Les quatre figures suivantes montrent, dans la région du volet, les nombres de Stanton et les coefficients de frottement pour les deux calculs. La zone de séparation mesure environ 1.1 10^{-2} m pour le cas de la soufflerie et 2 10^{-2} m pour le cas du vol. Un tourbillon secondaire peut être observé dans le cas du vol sur ligne charnière avec le volet. Enfin. nous montrons les courbes de convergence dans la région intermédiaire et dans celle du volet (figures 16 et 17).

6. Conclusion

Nous avons présenté les méthodes numériques utilisées dans le code CELHYO pour le calculs des écoulements en déséquilibre thermique et chimique. Ce code permet d'utiliser les schémas de Roe, d'Osher, de van Leer et leur hybridation. L'accent a été mis sur la technique de décentrement par hybridation. Cette technique de décentrement combine les deux approches classiques de manière à n'en retenir que les propriétés jugées favorables pour la simulation numérique d'écoulements où coexistent d'importants phénomènes non linéaires et linéaires. C'est en particulier le cas des écoulements visqueux hyperenthalpiques présentés. L'opérateur implicite est construit sur les flux de van Leer et est inversé par une méthode itérative de type GMRES. Ce code permet de calculer des configurations variées bidimensionnelles et axisymétriques d'écoulements hyperenthalpiques, et ce en utilisant des schémas numériques précis et en obtenant une bonne convergence.

Remerciements
Ce travail a été en partie effectué avec le soutien financier de la DRET et l'ESA/CNES.

References

[1] Abgrall R., Fezoui L., Talandier J.: *Extension of Osher's Riemann Solver to Chemical and Vibrational Nonequilibrium Gas Flows*, INRIA Report 1221, May 1990.

[2] Armaly B. F., Sutton K. : *Viscosity of Multicomponent Partially Ionized Gas Mixtures* , AIAA-80-1495, AIAA 15th Thermophysics Conference, 1980.

[3] Blottner F. G., Johnson M., Ellis M. : *Chemically Reacting Viscous Flow Program For Multicomponent Gas Mixtures* , Sandia Laboratories Report 87115, 1971.

[4] Coquel F., Flament C., Joly V., Marmignon C. : *Viscous Nonequilibrium Flow Calculations* , Computing Hypersonic Flows, Volume 3, ed. Bertin J.J., Périaux J., Ballmann J., Birkhäuser, Boston, 1993.

[5] Coquel F., Joly V. : *Développement d'un Code de Calcul d'Ecoulements Hypersoniques en Déséquilibre Chimique et Vibrationnel*, Rapport interne ONERA non publié, Juillet 1991.

[6] Coquel F., Joly V. : *Développement d'un Code de Calcul d'Ecoulements Hypersoniques en Déséquilibre Chimique et Vibrationnel*, Rapport interne ONERA non publié, Octobre 1992.

[7] Coquel F., Liou M.S. : *Field by Field Hybrid Upwind Splitting Methods*, AIAA-93-3302-CP, 1993.

[8] Coquel F., Liou M.S. : *Hybrid Upwind Splitting (HUS) by a Field by Field Decomposition*, NASA TM 106843, ICOMP-95-2, 1995.

[9] Gardiner W.C.Jr.: *Combustion Chemistry*, Springer Verlag 1984

[10] Larrouturou B. : *On Upwind Approximations of Multi-dimensional Multi-species Flows*, Computational methods in applied sciences, Ch. Hirsch (Eds), Elsevier, 1992.

[11] Liou M.S., Van Leer B. and Shuen J.S.: *Splitting of Inviscid Fluxes for Real Gases*, J. Comp. Phys., Vol. 81, pp.1-24, March 1990.

[12] Marmignon C., Joly V. et Coquel F.: *Calculs d'Ecoulements Visqueux en Déséquilibre dans des Tuyères*, Agard Conference Proceedings 514, 1992.

[13] Mehlman G. : *An Approximate Riemann Solver for Fluids Systems Based on a Shock-Curve Decomposition*, Third International Conference on Hyperbolic Problem, Uppsala, Suède, 1993.

[14] Millikan R. C., White D. R. : *Systematics of Vibrational Relaxation* , J. Chem. Phys., Vol.36, No.12, 1963.

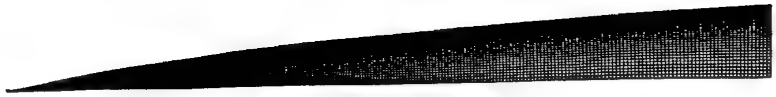


Fig. 1 – Maillage de la tuyère

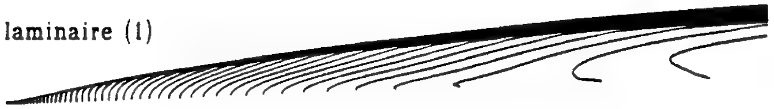


Fig. 2 – Champ des nombres de Mach (pas =0,25)

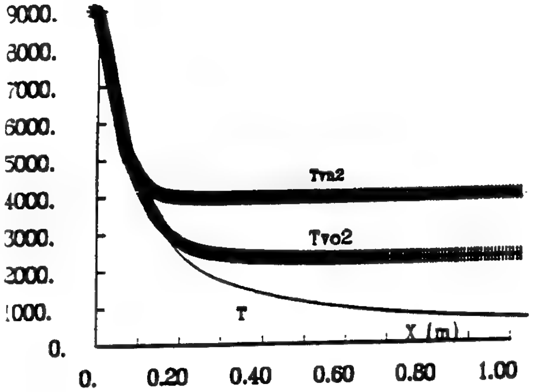


Fig. 3 – Distributions des températures le long de l'axe

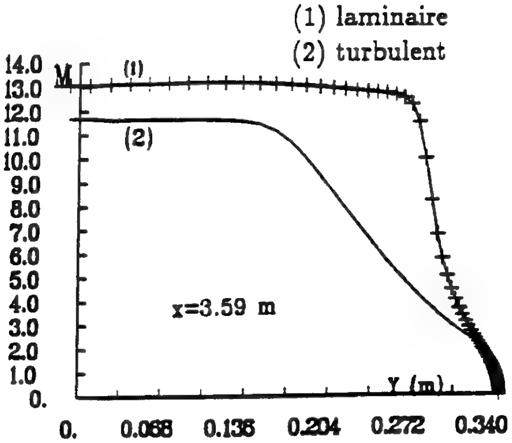


Fig. 4 – Distributions des nombres de Mach.

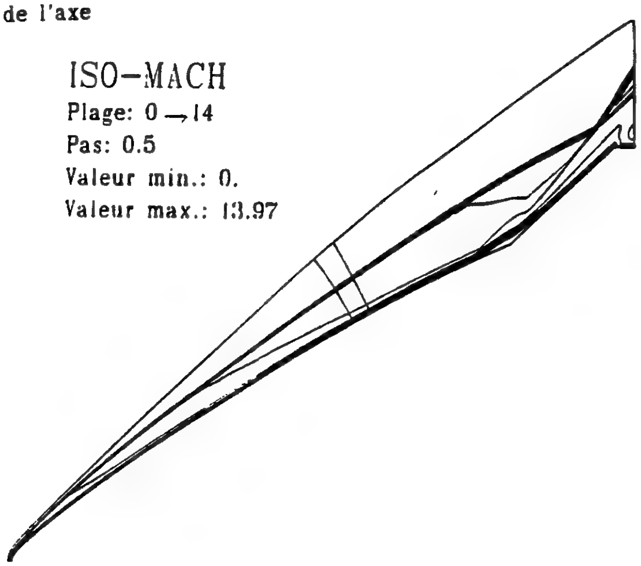


Fig. 5 – Champ des nombres de Mach (cas tuyère)

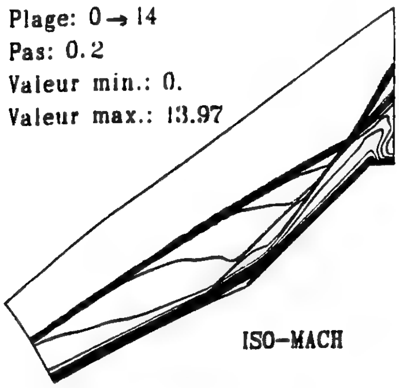


Fig. 6 – Champ des nombres de Mach (volet)

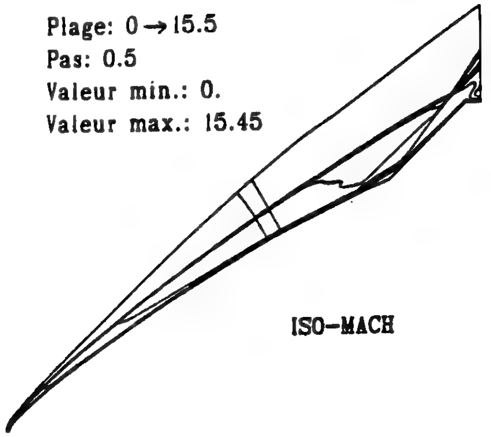


Fig. 7 – Champ des nombres de Mach (cas vol)

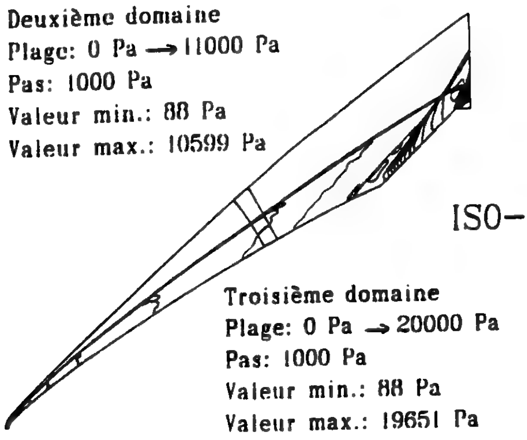


Fig. 8 – Champ des pressions (cas tuyère)

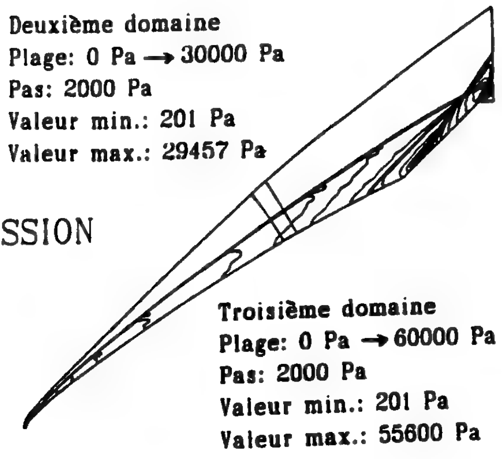


Fig. 9 – Champ des pressions (cas vol)

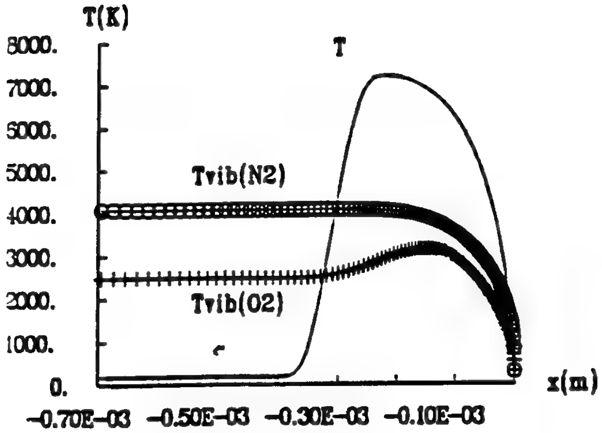


Fig. 10 - Distributions des températures sur l'axe

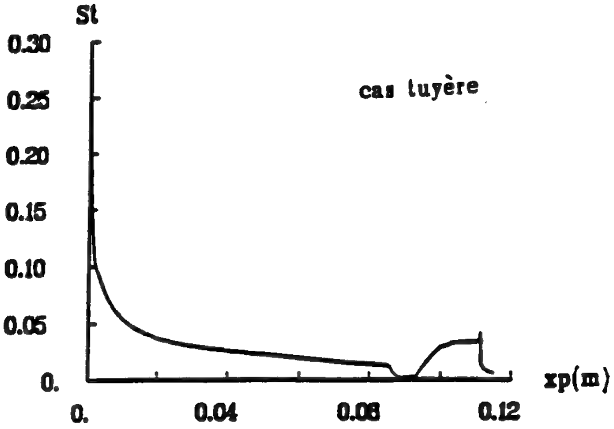


Fig. 11 – Distribution des nombres de Stanton à la paroi

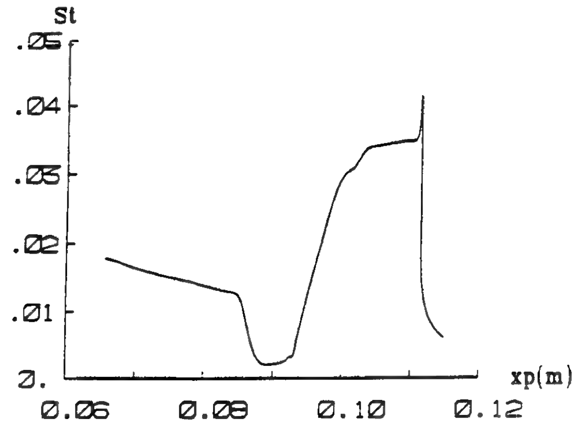


Fig. 12 – Distribution des nombres de Stanton
cas tuyère, volet

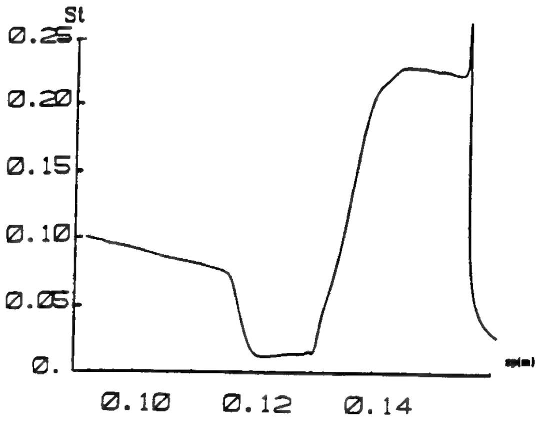


Fig. 13 – Distribution des nombres de Stanton
cas vol, volet

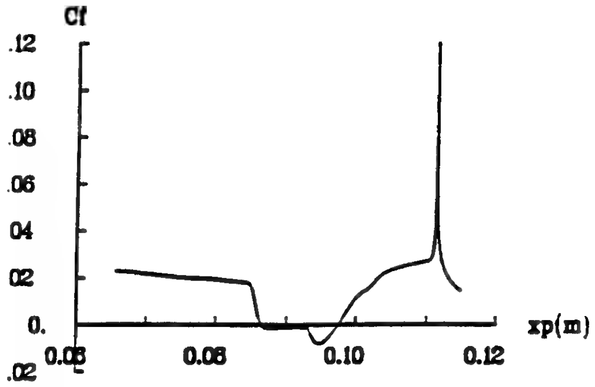


Fig. 14 - Distribution des coefficients de friction
cas tuyère, volet

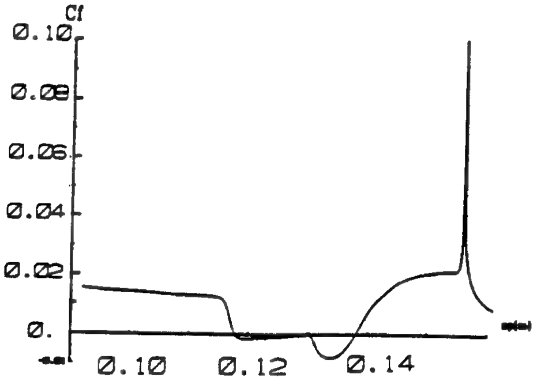


Fig. 15 – Distribution des coefficients de friction
cas vol, volet

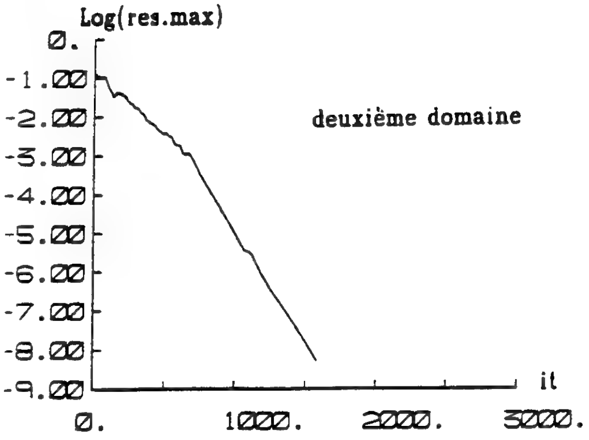


Fig. 16 – Courbe de décroissance des résidus

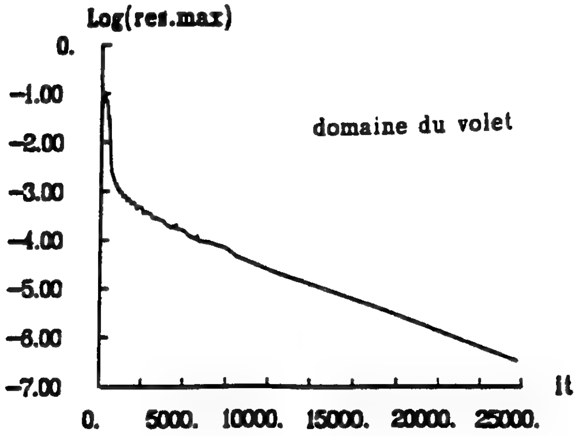


Fig. 17 – Courbe de décroissance des résidus

A PROJECTION METHODOLOGY FOR THE SIMULATION OF UNSTEADY INCOMPRESSIBLE VISCOUS FLOWS USING THE APPROXIMATE FACTORIZATION TECHNIQUE

A. Pentaris
S. Tsangaris

Laboratory of Aerodynamics, National Technical University of Athens
PO Box 64070, 15710 Zografou, Athens, Greece

SUMMARY

In this paper, an implicit projection methodology for the solution of the two-dimensional, time dependent, incompressible Navier - Stokes equations is presented. The basic principle of this method is that the evaluation of the time evolution is split into intermediate steps. The computational method is based on the approximate factorization technique. The coupled approach is used to link the equations of motion and the turbulence model equations. The standard k-ε turbulence model is used. The current methodology, which has been tested extensively for steady problems, is now applied for the numerical simulation of unsteady flows. Several cases were tested, such as plane or axisymmetric channels, a backward facing step and a flow behind a square cylinder.

1. INTRODUCTION

The numerical prediction of unsteady incompressible flowfields has always been one of the most challenging areas of fluid dynamics. The primary difficulty is in finding a satisfactory way to link changes in the velocity fields to changes in the pressure field. This interaction must be accomplished in such a manner as to ensure that the divergence of the velocity vanishes at each level of physical time. The most common solution to this problem is the use of an artificial compressibility methodology or a projection methodology.

The projection method for the solution of the time-dependent Navier-Stokes equations was introduced independently by Chorin (Ref 1) and Temam (Ref 2). Subsequently, an explicit version of the method was presented by Fortin et al (Ref 3). The projection method is an interpretation of a fractional-step method as adapted to the unsteady Navier-Stokes equations (Ref 4).

The procedure of the physical time level increment is split into two steps. Following the decomposition of Chorin (Ref 1), a tentative velocity field is first calculated by the discretized momentum equations without the pressure gradient. At the second step, the velocity components at the new time level are evaluated by correcting the tentative solution in order to satisfy the incompressibility constraint.

The solution algorithm we use in the present study, is the approximate factorization technique. This is an implicit algorithm which was initially developed by Beam and Warming (Ref 5) for compressible flows but has successfully used for incompressible steady flows as well (Ref 6, 7). Regarding the mathematical model, a projection method is developed, which uses a Poisson equation for the explicit pressure derivation, while the numerical algorithm involves only the momentum equations.

Concerning the turbulence model there are plenty of options. The standard k-ε model with the wall functions equations (Ref 8) was selected because it is well tested and widely used, in spite of its disadvantages. In addition, small values of the y-plus are not required, so coarse grids can be used near the walls and thus large time steps are possible. It is expected, that this turbulence model will sometimes perform poorly, especially in the recirculation zones.

The objective of this paper is to describe a new projection methodology developed for collocated grids and to present predictions for several test cases where the unsteadiness is either forced or inherent.

2. THE GOVERNING EQUATIONS

The full form of the momentum equations is used, where all variables are in non-dimensional form. Concerning the turbulent flows the high-Reynolds number (Ref 8) form of the k-ε model is used.

This formulation requires the use of the wall functions to bridge the viscous and boundary layers in proximity to the solid wall. This approach is strictly valid only for attached shear layers and may perform poorly in the recirculation zones. In addition this model is valid under the hypothesis of equilibrium and may not satisfactory perform in unsteady flows.

On the other hand, experimental observations showed that the general behaviour of the boundary layer and the structure of the turbulence are not fundamentally affected by the unsteadiness of the flow (Ref 9, 10, 11). From these observations it is well founded to suppose that the hypotheses used in calculations methods for the steady case are still valid for the unsteady case.

The reference quantities are some reference velocity u_{ref}, a reference length L_{ref}, a reference density ρ_{ref} and a reference kinematic viscosity v_{ref}. The reference value for the time is defined as $t_{ref}= L_{ref}/u_{ref}$ and for the pressure is the prod-

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

uct $p_{ref}=\rho_{ref}u^2_{ref}$. The reference quantity for the turbulent kinetic energy is u^2_{ref} and for the dissipation rate u^3_{ref}/L_{ref}.

Performing a generalised coordinates' transformation from the physical (x,y,t) to the computational (ξ,η,τ) domain, the following non-dimensional form of the equations is obtained (Ref 12):

$$\partial_\tau Q + \partial_\xi F + \partial_\eta G + \alpha E + K = \partial_\xi V + \partial_\eta W + \alpha C + D$$

where is $\alpha=0$ for the two dimensional equations, $\alpha=1$ for the axisymmetric equations and the subscripts x,y,ξ,η,τ denote derivation. For convenience we express the above equation in the following form:

$$\partial_\tau Q + K = [\mathcal{F}(u, v)] \tag{1}$$

where

$$[\mathcal{F}(u, v)] = \partial_\xi V + \partial_\eta W + \alpha C + D - \partial_\xi F - \partial_\eta G - \alpha E$$

In equation (1), Q is the vector of the conservative variables:

$$Q = \frac{1}{J}[u, v, k, \varepsilon]^T$$

F,G,E are the convective fluxes:

$$F = \frac{1}{J}\left[u\,U + \frac{2}{3}\xi_x k,\ v\,U + \frac{2}{3}\xi_y k,\ k\,U,\ \varepsilon\,U\right]^T$$

$$G = \frac{1}{J}\left[u\,V + \frac{2}{3}\eta_x k,\ v\,V + \frac{2}{3}\eta_y k,\ k\,V,\ \varepsilon\,V\right]^T$$

$$E = \frac{v}{Jy}[u, v, k, \varepsilon]^T$$

V,W,C are the viscous fluxes:

$$V = \frac{1}{J\,Re}\begin{pmatrix} \xi_x\tau_{xx} + \xi_y\tau_{xy} \\ \xi_x\tau_{xy} + \xi_y\tau_{yy} \\ \Gamma_k(\xi_x k_x + \xi_y k_y) \\ \Gamma_\varepsilon(\xi_x\varepsilon_x + \xi_y\varepsilon_y) \end{pmatrix} W = \frac{1}{J\,Re}\begin{pmatrix} \eta_x\tau_{xx} + \eta_y\tau_{xy} \\ \eta_x\tau_{xy} + \eta_y\tau_{yy} \\ \Gamma_k(\eta_x k_x + \eta_y k_y) \\ \Gamma_\varepsilon(\eta_x\varepsilon_x + \eta_y\varepsilon_y) \end{pmatrix}$$

$$C = \frac{1}{J\,Re\,y}\begin{pmatrix} \tau_{xy} \\ \tau_{yy} - \tau_{\varphi\varphi} \\ \Gamma_k k_y + 2\nu_t \frac{v^2}{y} \\ \Gamma_\varepsilon \varepsilon_y + 2\nu_t C_1 \frac{\varepsilon}{k}\frac{v^2}{y} \end{pmatrix}$$

D is a vector that contains the source terms of the k and ε equations:

$$D = \frac{1}{J}\left[0,0,\ C_\mu \frac{k^2}{\varepsilon}\tilde{G} - \varepsilon,\ C_\mu C_1 k\tilde{G} - C_2\frac{\varepsilon^2}{k}\right]^T$$

and, finally

$$K = \begin{pmatrix} \partial_\xi\left(\xi_x \frac{p}{J\rho}\right) + \partial_\eta\left(\eta_x \frac{p}{J\rho}\right) \\ \partial_\xi\left(\xi_y \frac{p}{J\rho}\right) + \partial_\eta\left(\eta_y \frac{p}{J\rho}\right) \\ 0 \\ 0 \end{pmatrix} = \frac{1}{J}\begin{pmatrix} \partial_x\left(\frac{p}{\rho}\right) \\ \partial_y\left(\frac{p}{\rho}\right) \\ 0 \\ 0 \end{pmatrix}$$

is a matrix that contains the pressure derivatives of the momentum equations.

In the expressions above, ξ,η are the curvilinear coordinates, connected to the cartesian ones x,y through the generalised coordinates' transformation:

$$\xi = \xi(x, y, t)\ ,\ \eta = \eta(x, y, t)\ ,\ \tau = t$$

and J is the Jacobian of the transformation:

$$J = \xi_x\eta_y - \xi_y\eta_x$$

In addition, U, V are the contravariant velocities along the ξ,η directions respectively, given by the following relations:

$$U = \xi_t + \xi_x u + \xi_y v \quad , \quad V = \eta_t + \eta_x u + \eta_y v$$

Re is the Reynolds number and $\tilde{G}$ is the kinetic energy production term:

$$\tilde{G} = 2\left[(u_x)^2 + (v_y)^2\right] + (u_y + v_x)^2$$

The stresses are:

$$\tau_{xx} = 2\nu_{eff}u_x\ ,\ \tau_{yy} = 2\nu_{eff}v_y\ ,\ \tau_{\varphi\varphi} = 2\nu_{eff}v / y$$

$$\tau_{xy} = \tau_{yx} = \nu_{eff}(u_y + v_x)$$

where ν_{eff} is the effective viscosity. Finally, for the turbulence model equations are:

$$\Gamma_k = \nu_l + \frac{\nu_t}{\sigma_k} \quad , \quad \Gamma_\varepsilon = \nu_l + \frac{\nu_t}{\sigma_\varepsilon}$$

where ν_l is the kinematic viscosity and ν_t is the turbulent viscosity, which is given by the relation:

$$\nu_t = Re\ C_\mu \frac{k^2}{\varepsilon}$$

The constants are:

$$C_\mu = 0.09, C_1 = 1.44, C_2 = 1.92, \sigma_k = 1.0, \sigma_\varepsilon = 1.3$$

For the above model the concept of wall functions has been employed. The central idea is that the flow in the region near the wall can be assumed to behave as an one-dimensional Couette flow. This is a reasonable assumption except for regions of high pressure gradient, separation or reattachment. Once this assumption is made, it is rather easy to arrive at exact or semi-empirical relations (Ref 8, 14), which link the shear stresses and the other variables at the wall to the values of velocity, turbulence energy, etc. at the outer edge of the Couette layer, where the first interior grid point is located.

3. NUMERICAL ALGORITHM

The time marching scheme

For the solution of the system of equation

(1) the implicit, factored, finite difference scheme of Beam and Warming (Ref 5) is used. The temporal derivative in equation (1) is approximated via a generalized time differencing:

$$Q_\tau^n = \frac{1}{\Delta\tau}\,\frac{(1+\zeta)\Delta - \zeta\,\nabla}{1+\theta\,\Delta}\,Q^n + O\left[\left(\theta - \frac{1}{2} - \zeta\right)\Delta\tau + \Delta\tau^2\right]$$

which takes the form:

$$\frac{\Delta Q^n}{\Delta\tau} = \frac{\theta}{1+\zeta}\,\partial_\tau Q^{n+1} + \frac{1-\theta}{1+\zeta}\,\partial_\tau Q^n + \frac{\zeta}{1+\zeta}\,\frac{\Delta Q^{n-1}}{\Delta\tau} + O\left[\left(\theta - \frac{1}{2} - \zeta\right)\Delta\tau + \Delta\tau^2\right] \tag{2}$$

where Δ and ∇ are the forward and backward differencing operators, respectively, the superscript n denotes the time instant and O denotes the order of the truncation error.

After substituting (1) into (2) and performing calculations the following relation is derived:

$$\frac{Q^{n+1} - Q^n}{\Delta\tau} = \frac{\theta}{1+\zeta}\,[F(u,v)]^{n+1} + \frac{1-\theta}{1+\zeta}\,[F(u,v)]^n - \frac{\theta}{1+\zeta}\,K^{n+1} - \frac{1-\theta}{1+\zeta}\,K^n + \frac{\zeta}{1+\zeta}\,\frac{\Delta Q^{n-1}}{\Delta\tau} + O\left[\left(\theta - \frac{1}{2} - \zeta\right)\Delta\tau + \Delta\tau^2\right]$$

Using a fractional step method similar to that described by Anderson and Kristoffersen (Ref 13) the above relation is split in two parts:

$$\frac{Q^* - Q^n}{\Delta\tau} = \frac{\theta}{1+\zeta}\,[F(u,v)]^* + \frac{1-\theta}{1+\zeta}\,[F(u,v)]^n + \frac{\zeta}{1+\zeta}\,\frac{\Delta Q^{n-1}}{\Delta\tau} + O\left[\left(\theta - \frac{1}{2} - \zeta\right)\Delta\tau + \Delta\tau^2\right] \tag{3}$$

and

$$\frac{Q^{n+1} - Q^*}{\Delta\tau} = \frac{\theta}{1+\zeta}\,[F(u,v)]^{n+1} - \frac{\theta}{1+\zeta}\,[F(u,v)]^* - \frac{\theta}{1+\zeta}\,K^{n+1} - \frac{1-\theta}{1+\zeta}\,K^n$$

where Q^* is an intermediate, or tentative, flowfield. Using equation (1) the above relation is written in the form:

$$\frac{Q^{n+1} - Q^*}{\Delta\tau} = \frac{\theta}{1+\zeta}\,\partial_\tau Q^{n+1} - \frac{\theta}{1+\zeta}\,\partial_\tau Q^* - \frac{\theta}{1+\zeta}\,K^* - \frac{1-\theta}{1+\zeta}\,K^n$$

and after some simple calculations and assuming $K^{n+1}=K^*$ is obtained:

$$\frac{Q^{n+1} - Q^*}{\Delta\tau} = -\frac{\theta}{1+\zeta-\theta}\,K^{n+1} - \frac{1-\theta}{1+\zeta-\theta}\,K^n \tag{4}$$

Equation (4) imposes the condition $1+\zeta-\theta\neq0$. Thus we use $\theta=1$ and $\zeta=0.5$ which leads to the second order three point backward scheme. Equation (3) is actually the same with (2), except that it contains equation (1) without the pressure gradients, and is

$$\Delta Q^n = Q^* - Q^n$$

A non-linear expression, eq. (3), occurs for the time increment of the conservatives variables' vector ΔQ^n (Ref 12, 14). In order to derive a linear algebraic system of equations, a linearization of viscous and inviscid fluxes must be performed. The inviscid fluxes, which are functions of Q, are linearized using a Taylor series expansion, for example:

$$\Delta F^n = A^n \cdot \Delta Q^n + O\left(\Delta\tau^2\right)$$

where $A^n=\partial F^n/\partial Q^n$ is the Jacobian matrix of the vector F^n.

The above linearization of the inviscid fluxes ensures the second order time accuracy of the scheme. In order that this accuracy is retained in the corresponding linearization of the viscous fluxes, it must be taken into account that the latter are functions of all Q, Q_ξ, Q_η, for example:

$$V^n(Q, Q_\xi, Q_\eta) = V_1^n(Q, Q_\xi) + V_2^n(Q, Q_\eta)$$

The linearization of matrix V_1^n leads to the following relation:

$$\Delta V_1^n = -\left(-P^n + R_\xi^n\right)\Delta Q^n + \left(R^n\,\Delta Q^n\right)_\xi + O\left(\Delta\tau^2\right)$$

while the matrix V_2^n is treated in a explicit way:

$$\Delta V_2^n = \Delta V_2^{n-1} + O\left(\Delta\tau^2\right)$$

where P^n and R^n are the Jacobian matrices. A detailed description for all the linearizations is given in Ref 14.

The substitution of the linear expressions of the flux vectors into the original nonlinear equation for ΔQ^n, leads to a strongly coupled system of equations in both spatial directions. This coupled system is solved by the Approximate Factorization Technique (Ref 5, 14), which leads to the following two tridiagonal systems, one for each of the two directions ξ,η:

$$\left\{I + \frac{\theta\Delta\tau}{1+\zeta}\left[\partial_\xi\left(A - P + R_\xi\right) - \partial_{\xi\xi}R - \alpha N_1 + \Theta_a H\right]^n\right\} \cdot \Delta\tilde{Q}^n = \text{R. H. S.} \tag{5a}$$

$$\left\{I + \frac{\theta\Delta\tau}{1+\zeta}\left[\partial_\eta\left(B - Y + S_\eta\right) - \partial_{\eta\eta}S - \alpha\left(N_2 + N_3 - T\right) + \Theta_b H\right]^n\right\} \cdot \Delta Q^n = \Delta\tilde{Q}^n \tag{5b}$$

where $A, B, P, Y, R, S, N_1, N_2, N_3, T$ and H are Jacobian matrices (Ref 14), and

$$\overline{Q}^* = \overline{Q}^n + J \cdot \Delta Q^n \tag{5c}$$

R. H. S. =

$$\frac{\Delta\tau}{1+\zeta}\left[\partial_\xi(-F+V)^n+\partial_\eta(-G+W)^n+\alpha(C-E)^n+D^n\right]$$
$$+\frac{\theta\,\Delta\tau}{1+\zeta}\left[\partial_\xi\Delta V_2^{n-1}+\partial_\eta\Delta W_1^{n-1}\right]$$
$$+\frac{\zeta}{1+\zeta}\Delta Q^{n-1}+D_e+O\left[\left(\theta-\frac{1}{2}-\zeta\right)\Delta\tau^2+\Delta\tau^3\right] \quad (5d)$$

where $\overline{Q} = JQ$ is the vector of conservative variables in the physical domain, D_e is the artificial dissipation terms (Ref 14), and Θ_a, Θ_b are weighting functions (Ref 15) used to add the Jacobian matrix H in both the sweeps.

The Poisson equation

Equation (4) leads to the following relations (θ=1):

$$u^{n+1} = u^* - \Delta\tau\frac{\theta}{1+\zeta-\theta}\partial_x\left(\frac{p}{\rho}\right)^{n+1} \quad (6a)$$

$$v^{n+1} = v^* - \Delta\tau\frac{\theta}{1+\zeta-\theta}\partial_y\left(\frac{p}{\rho}\right)^{n+1} \quad (6b)$$

$$k^{n+1} = k^*,\ \ \varepsilon^{n+1} = \varepsilon^* \quad (6c)$$

Assuming that the continuity equation is satisfied at the n+1 time instant:

$$\nabla\cdot\vec{c}^{\,n+1} = 0$$

the first two of (6) are combined to give the Poisson equation:

$$\nabla^2\left(\frac{p}{\rho}\right)^{n+1} = \frac{1+\zeta-\theta}{\theta\,\Delta\tau}\nabla\cdot\vec{c}^{\,*}$$
$$-\frac{1}{\Delta\tau}\left[\partial_x\Delta\tau\cdot\partial_x\left(\frac{p}{\rho}\right)^{n+1}+\partial_y\Delta\tau\cdot\partial_y\left(\frac{p}{\rho}\right)^{n+1}\right] \quad (7)$$

where $\vec{c} = (u, v)$ is the velocity vector.

The procedure that is used is the following. First the time-marching scheme of (5) is solved to provide the tentative velocity components u^*, v^* and the turbulent variables k,ε. Next the Poisson equation (7) is solved using the classic ADI method and the pressure field is obtained. Finally the velocity components at the new time level are evaluated by correcting the tentative velocity field using (6a) and (6b). It is essential, for unsteady flows, to fully converge the Poisson equation at each time step in order the mass conservation to be satisfied.

The artificial dissipation terms

The spatial derivatives in the above system of equations are approximated by three point central second order differencing expressions. So the solution of the system of equations (5) requires the inversion of two block tridiagonal systems, one in each direction. On the other hand, the use of central differences on collocated grids leads to the necessity of adding external artificial dissipation terms, so that the stability is retained and high frequency oscillations from the solution are removed. In the present work only explicit terms D_e are used in (5). These terms are a blended second and fourth order non-linear model which is widely used in compressible flows (Ref 16, 17, 18, 19) and was used for the first time in incompressible flows by Pentaris et al (Ref 14), where is proved that the existence of the second order dissipation terms do not affect the spatial accuracy of the method.

The definition of the time step

Although the solution method is implicit, the actual stability of the scheme is not independent of the time step used. In this work small time steps are used which help the fast convergence of the Poisson equation. When a problem with oscillating flow rate is to be simulated, the Navier-Stokes equations must be integrated for as many cycles as are needed to reach a periodic steady state, if such a state exists. In the periodic steady state, of period T, the solutions at time instants t and t+T must reach a specified convergence criterion, which in the present work is 1×10^{-4}. With the present method this criterion is reached at the second period, because 10000 time intervals are used per period. Using less time intervals per period, more iterations are needed for the convergence of the Poisson equation. In addition more periods are necessary to reach the above criterion and thus the total computational cost is increased.

When a problem with steady upstream conditions is solved, where the Poisson equation is rapidly converged, the time step is essential to be as large as possible. Then the time step is defined as:

$$dt = \frac{CFL}{1+\sqrt{J_{max}}}$$

where J_{max} is the maximum of all the Jacobians in the computational domain and CFL is the Courant number.

4. BOUNDARY CONDITIONS

The use of a collocated grid allows the impose of the suitable boundary conditions in convenient form. Throughout the computations, explicit boundary conditions are used. For the Poisson equation these conditions are derived by integrating equation (7) over the solution domain and applying the Gauss's theorem (Ref 20):

$$\oint_A \nabla\left(\frac{p}{\rho}\right)^{n+1}\cdot\vec{n}\ dA = \frac{1+\zeta-\theta}{\theta\,\Delta\tau}\oint_A \vec{c}^{\,*}\cdot\vec{n}\ dA$$

where the last part of equation (7) vanish for unsteady flows because is Δτ=const in the entire domain. In the equation above, $\vec{n}$ is the outward unit vector normal to the boundary A which encloses the solution domain.

Concerning the other variables, the velocity profiles u_{in} are specified in the inlet boundary , while the kinetic energy k_{in} and dissipation rate ε_{in} are given by the following relations:

$$k_{in} = 0.003\, u_{in}^2 \ , \ \varepsilon_{in} = \frac{C_\mu\ k_{in}^2}{0.005\ \ D_{in}} \qquad (8)$$

where D_{in} is the inlet span.

On the outlet boundary all variables are calculated by extrapolation from the interior. At the symmetry axis the first derivatives of all variables are set equal to zero, except the v-component of the velocity which is set equal to zero. On the solid surface the non-slip condition is applied for the velocity components. The kinetic energy and the dissipation rate are defined at the first grid point above the solid surface with the use of the wall functions (Ref 14).

Finally, as initial conditions, the u velocity component is set equal to unity, while the v velocity component and the pressure vanish. The initial data for the turbulence model variables are given by equations (8).

5. RESULTS AND VALIDATION

Some representative results of several test cases are shown in this section. It must be mentioned that all the quantities used are dimensionless. The dimensionless numbers Reynolds, Strouhal and Womersley are defined as:

$$Re = \frac{u_{ref}\ L_{ref}}{\nu_{ref}} \ , \ Str = \frac{\omega_{ref}\ L_{ref}}{u_{ref}} .$$

$$W = L_{ref}\sqrt{\frac{\omega_{ref}}{\nu_{ref}}} = \sqrt{Str \cdot Re}$$

respectively, where ω_{ref} is the reference cyclic frequency.

Finally it must be noted that all the results have been tested for various grids and are independent from the grid density.

One-dimensional oscillatory flow

In order to check the reliability of the present method it was initially developed for one-dimensional flows and it was tested to an oscillatory channel flow (Ref 21). In this problem the back pressure of the channel is oscillating according to:

$$p_{ex}(t) = p_o + p_e \sin\left(Str \cdot t\right)$$

An analytic solution to this problem can only be obtained if the pressure perturbation p_e is small compared to the mean back pressure p_o. In this work these parameters are p_e=0.1 and p_o=1. The Strouhal number, based on the time mean inflow velocity u_o and the channel length l, $Str=\omega_{ref}l/u_o$ is chosen to be equal to 10.

The analytic solution for the velocity and for the pressure are given in Ref 21. These solutions show that the velocity is a function of time only. This is a direct reflection of the incompressible continuity equation in a constant area tube. The pressure fluctuation is a linear function of x that vanishes at x=1 to meet the downstream boundary condition. Some comparisons between numerical results and the analytic solution are shown in Fig 1. The calculated dimensionless velocity as a function of time, and the dimensionless pressure at three longitudinal positions of the tube are compared to the analytic solution. Both the numerical results are in excellent agreement with the analytic solution, demonstrating the reliability of the present method for unsteady flows.

Two-dimensional periodic flow between parallel plates

The oscillatory flow between two parallel plates with a span of 2b is the second test case we present. The Reynolds number is based on the half distance b between the two plates and the maximum inflow velocity u_o. At x=0 the imposed inflow uniform velocity is given by:

$u(t)=1\cdot\sin(Str\cdot t)$, $v(t)=0$

The analytic solution for the velocity and the pressure gradient for the developed part of the channel, is given by Moore (Ref 22). The Strouhal number is equal to 10 and the Reynolds number is equal to 1.6.

A 75×29 grid is used for the current test case, with 4b length and 1b height. The lower boundary is a solid wall and the upper one is a symmetry axis.

One cycle of the inflow velocity oscillation is split in 10000 time intervals and the dimensionless time step obtained is:

$$dt = \frac{2\pi}{Str \cdot 10000} = 2\pi \cdot 10^{-5}$$

In Fig 2 the developed velocity profiles at different physical time instants are presented. As can be seen the numerical results coincide with the analytic solution. In Fig 3 the velocity as a function of time at three different distances from the wall, and the pressure gradient in the developed part as a function of time are presented. The agreement is excellent comparing the numerical results with the analytic solution. It is clear that the unsteady motion is predicted well after the one fourth of the first period, and this is one reason for the use of small time steps.

Periodic flow in axisymmetric channel

The third test case under consideration is the periodic Stokes flow in a circular tube, extensively presented and analysed by many researchers (Ref 23, 24, 25, 26). In the present paper the Reynolds number, based on the radius α of the tube and the maximum inflow velocity u_o, is considered to be equal to 0.1, in order to approximate the Stokes flow. At x=0 the imposed velocity profile is (Ref 26):

$$u(t)=u(y)\cdot\cos(Str\cdot t)\ ,\ v(t)=0$$

where u(y) is equal to unity except the near the wall region were parabolically approaches zero. For the present case we select the typical Womersley number of $W=\alpha\sqrt{(\omega_{ref}/v_{ref})}=\sqrt{30}$ and the Strouhal number becomes $Str=\alpha\omega_{ref}/u_o=300$. The time step used is $2.094\cdot10^{-6}$.

A 45×40 grid is used, with 1.2α length and 1α height. The lower boundary is a solid wall and the upper one is a symmetry axis. Solution for the above relations are given by Goldberg et al (Ref 26), in their Table I.

In Fig 4 the comparisons between the semi-analytic solution and the numerical results provided by the current method are given, for the u-velocity component, at four instants of the physical time. The agreement of the current numerical results with the semi-analytic solution is very good at all the time instants. The discrepancies that occur at centreline velocity at ωt=0 and ωt=π due to the semi-analytic solution (Ref 26).

The main reason that this test case is examined, is that the results provided by the analytic solution concern the entire flowfield along the tube, in contrast to the flow between the two parallel plates where results only for the developed part of the flow were available. In addition the Strouhal number is much larger than it was in the previous test case.

Unsteady flow behind a square cylinder

The unsteady flow behind a square cylinder is presented in this paragraph. The objective is to examine the reliability of the methodology when the unsteadiness of the flow is due to the viscosity of the flow and not to an external cause.

The Reynolds numbers examined, based on the inflow uniform velocity u_o and the square side α, are 100, 250, 500 and 750. Three different grids were used with 100×56, 200×110 and 145×111 points. The 200×110 grid is shown in Fig 5. The points inside the square are blocked. The position of the cylinder and of all the boundaries are those shown in Fig 5, and are the same for all the grids. The upper and lower boundaries are considered to be symmetry axes.

Indicative experimental studies concerning this flow are those of Purtell and Klebanoff (Ref 27) and Okajima (Ref 28). Typical numerical studies are those of Davis and Moore (Ref 29), Franke et al (Ref 30) and Kelkar and Patankar (Ref 31).

In Fig 6 the Strouhal numbers $Str=f\alpha/u_o$ predicted for all the grids and for several time steps are shown. Comparisons are made to other experimental data and numerical results. The agreement is very good. It can be seen that the results are slightly affected by the grid density or the time step used. On the other hand, the disagreement between the experimental data presented in Fig 6 show the uncertainty and the sensitivity of the flow.

In Fig 7 the vorticity isolines are presented for Reynolds numbers 100 and 250. In Fig 8 the time history of the v-velocity behind the cylinder and the corresponding power spectrum are presented. It must be mentioned that for Reynolds numbers 100 and 250 the flow is periodic. For larger Reynolds numbers the flow becomes transitional or turbulent, and the time histories of the velocity and the pressure show a chaotic behaviour.

Unsteady turbulent flow behind a backward-facing step

In the present paper a numerical investigation of the coherent vortices in turbulence behind a backward-facing (Ref 32) step is presented. The ratio of the channel height W to the step height H is 2.5. The geometry and the inflow velocity profile U(y) are the same as in the experiments of Eaton and Johnston (Ref 33). A 250×50 grid is used, a detail of which is shown in Fig 9. The total length of the channel is 50 step heights. Both the lower and the upper boundaries are solid surfaces. The Reynolds number based upon the step height H and the maximum inflow velocity U_o is 38000. The time step used is 0.0075.

In the first run the original k-ε model was used. The flow that occurred was steady. The recirculation length was 7.1H. The main reason that a steady flow was predicted, is the overestimate of the turbulent viscosity, which indirectly reduces the Reynolds number. Thus a second run was performed using a modified relation for the turbulent viscosity:

$$\nu_t = C_\mu\, f_\mu\, \frac{k^2}{\varepsilon}$$

where

$$f_\mu = f_o + \left(1 - f_o\right)\left\{1 - \exp\left[-\left(y^+ - y_o^+\right) / A^+\right]\right\}^2$$

is a function proposed by Miner et al(Ref 34) in order to reduce the turbulent viscosity near the wall. The constants are $f_o = 0.04$, $y_o^+ = 8$ and $A^+=26$.

Using the above modification the flow becomes unsteady. The pressure contours and the vorticity contours are shown in Fig 10. The presence of a mixing layer behind the step is clear. The recirculation length (temporal mean) is overestimated and is 8.1H, versus the experimental result of 7.8H and the other numerical result of Silveira Neto et al (Ref 32) of 6.8H. The eddies which impinge on the lower wall, and are transported downstream, are shed with a frequency f that corresponds to a Strouhal number $Str=fH/U_o$ $=0.068$. This is in excellent agreement with the experimental data, where $Str \cong 0.07$.

In Fig 11 the time mean velocity profiles at two different positions are shown, in comparison to the experimental data of Eaton and Johnston and the numerical results of Silveira Neto et al. The agreement of the results provided with the experimental data is very good. At Fig 12 the time mean kinetic energy profiles are compared to the experimental data. The agreement is very good. In both the Fig 11 and 12 the results of the steady case are also shown. In Fig 13 the temporal evolution of the longitudinal velocity component at $x/H=7.59$, $y/H=0.1$ and the corresponding spectrum analysis are shown.

An interesting phenomenon, that can be observed in Fig 10 is the separation of the boundary layer from the upper wall; it generates a second street of coherent vortices which are transported toward the outlet of the channel with a Strouhal number $Str=0.068$. This phenomenon has also been observed in experiments performed by Armaly et al (Ref 35) with $Str \cong 0.07$.

6. CONCLUSIONS

An implicit projection methodology for the solution of the unsteady Navier-Stokes equations in collocated grids is presented in this paper. The computational method is based on the approximate factorization technique and the incompressibility constraint is satisfied by a Poisson equation. Extended comparisons with analytic solutions, experimental data and numerical results provided by other researchers lead to the conclusion that the present methodology is a reliable tool for solving a large range of unsteady problems.

7. REFERENCES

1. Chorin A. J., "Numerical solution of the Navier-Stokes equations", Math. Comp., vol. 22, pp. 745-762, 1968.
2. Temam R., "Sur l' approximation de la solution des equations de Navier-Stokes par la methode des pas fractionnaires (II)", Arch. Ration. Mech. and Anal., vol. 32, pp. 377-385, 1969.
3. Fortin M., Peyert R. and Temam T., "Resolution numerique des equations de Navier-Stokes pour un fluide incompressible", J. Mecan., .vel. 10, pp. 357-390, 1971.
4. Temam R., "Navier-Stokes equations", North-Holland, Amsterdam, 1979.
5. Beam R. M. and Warming R. F., "An implicit factored scheme for the compressible Navier-Stokes equations", AIAA Journal, vol. 16, No. 4, pp. 393-402, 1978.
6. Michelassi V. and Benocci C., "Prediction of incompressible flow separation with the approximate factorization technique", International Journal of Numerical Methods in Fluids, vol. 7, pp. 1383-1403, 1987.
7. Mansour M. L. and Hamed A., "Implicit solution of the incompressible Navier-Stokes equations on a non-staggered grid", Journal of Computational Physics, vol. 86, pp. 147-167, 1990.
8. Launder B. E. and Spalding D. B., "The numerical computation of turbulent flows", Computer Methods in Applied Mechanics and Engineering, vol. 3, pp. 269-289, 1974.
9. Zhong Q. and Olson M. D., "Periodic solution of turbulent oscillating channel flows", International Journal of Numerical Methods in Fluids, vol. 14, pp. 443-457, 1992.
10. Cousteix J., Desopper A. and Houdeville R., "Structure and development of a turbulent boundary layer in an oscillatory external flow", Turbulent Shear Flows I, pp. 154-171. Springer-Verlag, Berlin/Heidelberg/New York, 1977.
11. Binder G. and Kueny J. L., "Measurements of the periodic velocity oscillations near the wall in unsteady turbulent channel flow", Unsteady Turbulent Shear Flow, IUTAM Symp., pp. 100-108. Springer-Verlag, Berlin /Heidelberg/New York, 1977.
12. Tsangaris S., Thomadakis M. P. and Pentaris A., "Numerical investigation of axisymmetric compressible viscous flow", Proceedings of the 1st European Computational Fluid Dynamics, Elsevier pub., Brussels, Belgium, pp. 835-842, 1992.
13. Anderson H. I. and Kristoffersen R., "Numerical simulation of unsteady viscous flow", Arch. Mech., vol. 41, 2-3, pp. 207-223, 1989.
14. Pentaris A., Nikolados K. and Tsangaris S., "Development of projection and artificial compressibility methodologies using the approximate factorization technique", International Journal of Numerical Methods in Fluids, vol. 19, pp. 1013-1038, 1994.
15. Michelassi V. and Benocci C., "Efficient solution of turbulent incompressible separated flows", Proceedings of 8th GAMM Conference on Num. Methods in Fluid Mech. (1989), Vieweg Verlag, pp. 373-390, 1990.
16. Jameson A., Schmidt W. and Turkel E., "Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes", AIAA paper, No. 81-1259, 1981.
17. Pulliam T. H., "Artificial dissipation models for the Euler equations", AIAA Journal, vol. 24, No. 12, pp. 1931-1940, 1986.
18. Thomadakis M. P. and Tsangaris S., "Improved artificial dissipation schemes for the Euler equations", International Journal of Numerical Methods in Fluids, vol. 14,. pp. 1391-1405,

1992.

19. Dejean F., Vassilopoulos C., Simandirakis G., Giannakoglou K. C., Papailiou K. D., "Analysis of 2-D transonic turbomachinery flows using an explicit Low-Reynolds k-ε Navier-Stokes solver", Accepted for presentation at the 39th ASME Intl. Gas Turbine and Aeroengine Congress and Exposition, The Hague, June 13-16, 1994.
20. Sotiropoulos F. and Abdallah S., "Coupled fully implicit solution procedure for the steady incompressible Navier-Stokes equations", Journal of Computational Physics, vol. 87, pp. 328-348, 1990.
21. Merkle C. L. and Athavale M., "Time-accurate unsteady incompressible flow algorithms based on artificial compressibility", AIAA paper, No. 87-1137, 1987.
22. Moore F. K., "Theory of laminar flows", Princeton Univ. Press, Princeton, 1964.
23. Atabek H. B. and Chang C. C., "Oscillatory flow near the entry of a circular tube", Z. Angew. Math. Phys., vol. 12, pp. 185-201, 1961.
24. C. C. Chang and H. B. Atabek, "The inlet length for oscillatory flow and its effects on the determination of the rate of flow in arteries", Phys. Med. Biol., vol. 6, pp. 303-317, 1961.
25. Atabek H. B., Chang C. C. and Fingerson L. M., "Measurement of laminar oscillatory flow in the inlet length of a circular tube", Phys. Med. Biol., vol. 9, pp. 219-227, 1964.
26. Goldberg I. S., Carey G. F., McLay R. and Phinney L., "Periodic viscous flow: a bench-mark problem", International Journal of Numerical Methods in Fluids, vol. 11, pp. 87-97, 1990.
27. Purtell L. P. and Klebanoff P. S., "A low velocity airflow calibration and research facility", National Bureau of Standards Technical Note, No 989, 1979.
28. Okajima A., "Strouhal numbers of rectangular cylinders", Journal Fluid Mechanics, 123, 379-398, 1982.
29. Davis R. W. and Moore E. F., "An numerical study of vortex shedding from rectangles", Journal Fluid Mechanics, 116, 475-506, 1982.
30. Franke R. and Rodi W., "Calculation of vortex shedding past a square cylinder with various turbulence models", 8th Symposium for Turbulent Shear Flows, Technical University of Munich, 9-11 September, 1990.
31. Kelkar K. M. and Patankar S. V., "Numerical prediction of vortex shedding behind a square cylinder", International Journal for Numerical Methods in Fluids, 14, 327-341, 1992.
32. Silveira Neto A., Grand D., Metais O. and Lesiuer M., "A numerical investigation of the coherent vortices in turbulence behind a backward-facing step", J. Fluid Mech., vol. 256, pp. 1-25, 1993.
33. Eaton J. K. and Johnston J. P., "Turbulent flow re-attachment: an experimental study of the flow and structure behind a backward-facing step", Stanford University, Rep. Md-39, 1980.
34. Miner E. W., Swean T. F., Handler R. A. and Leighton R. I., "Examination of wall damping for the k-ε turbulence model using direct simulations of turbulent channel flow", International Journal for Numerical Methods in Fluids, 12, 609-624, 1991.
35. Armaly B. F., Durst F., Pereira J. C. and Schonung B., "Experimental and theoretical investigation of backward-facing step flow", J. Fluid Mech., vol. 127, pp. 423-496, 1983.

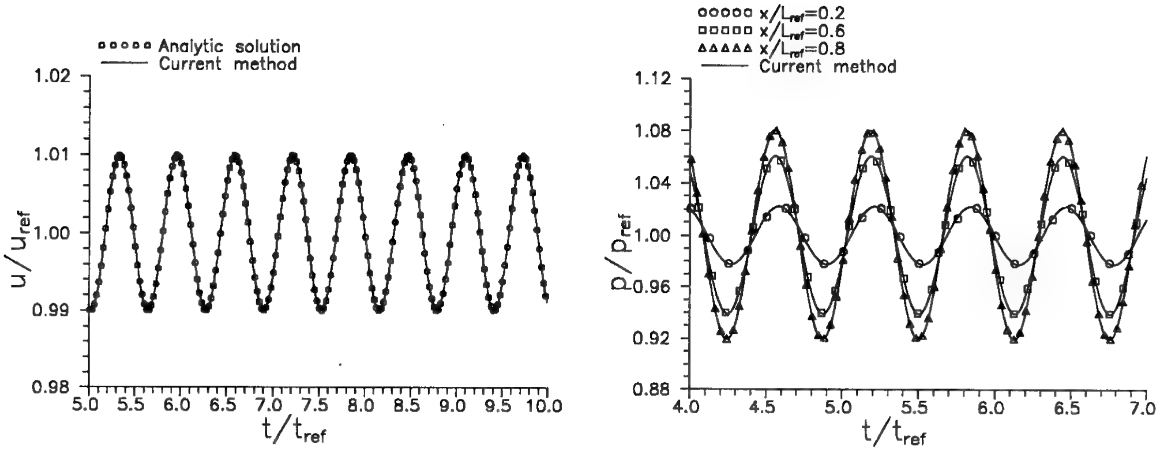


Figure 1. Time evolution of velocity (left) and pressure (right) in the one-dimensional flow. Comparison with analytic solution.

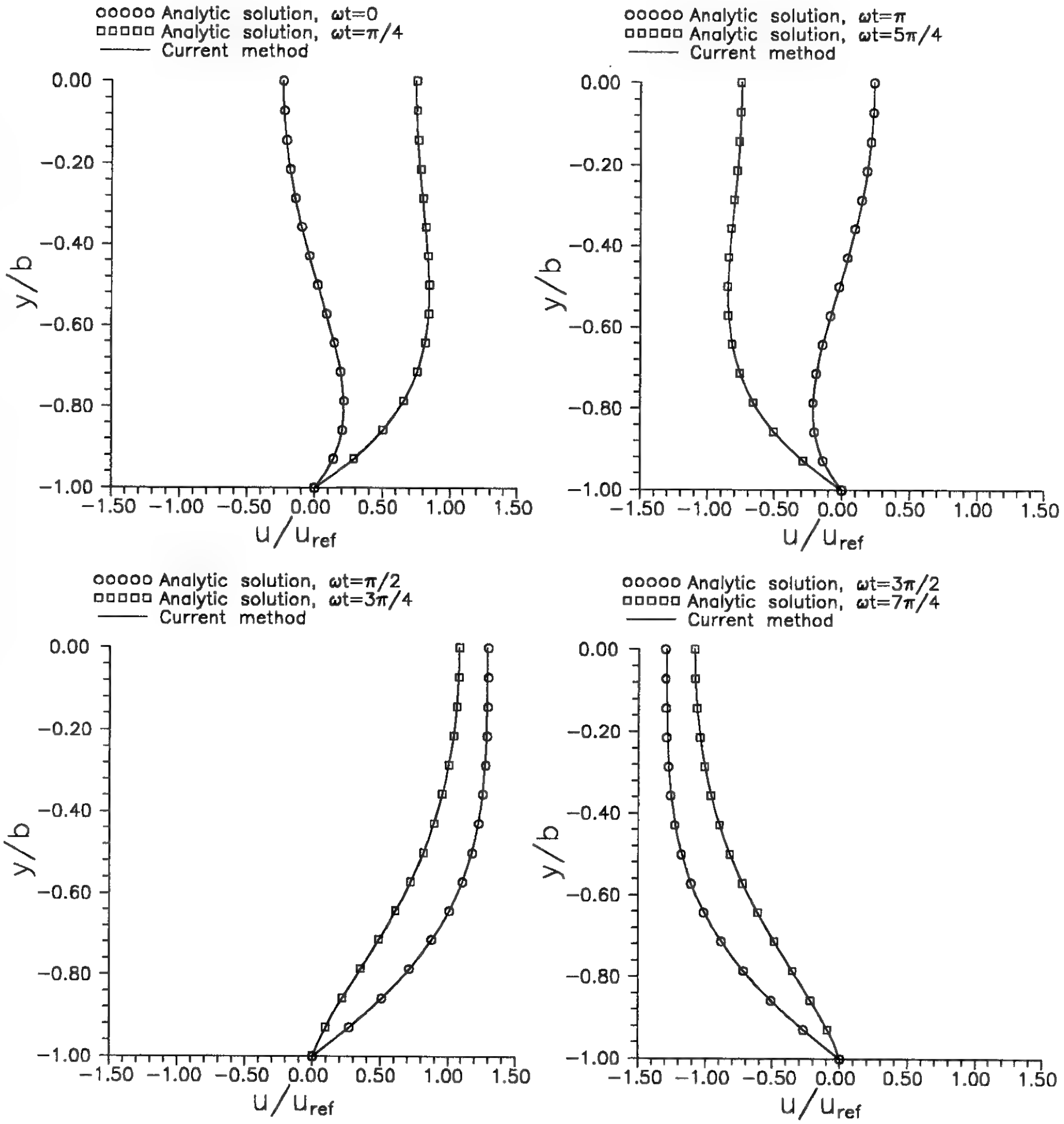


Figure 2. Longitudinal velocity profiles at several time instants, in the developed region of the two-dimensional channel.

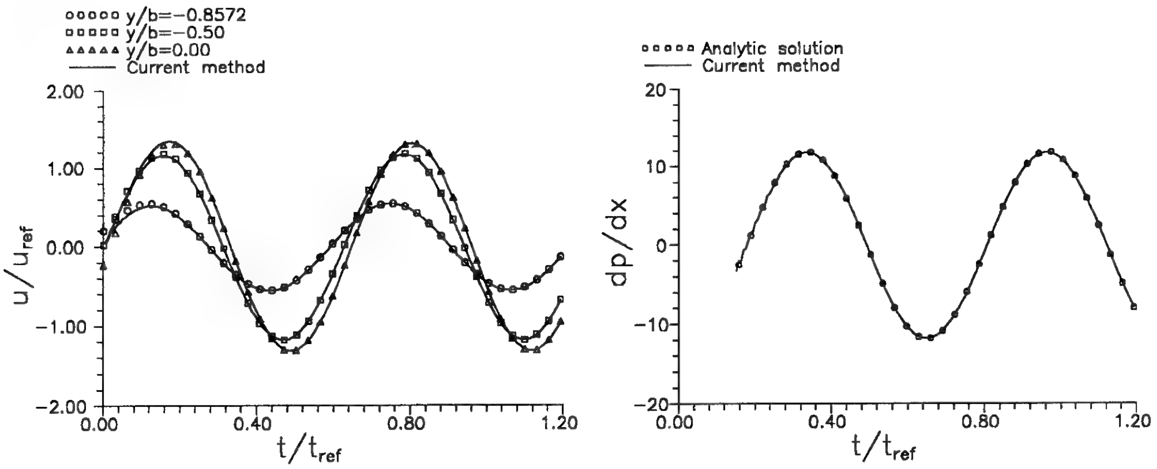


Figure 3. Time evolution of the velocity at three radial positions (left) and time evolution of the pressure gradient (right) in the developed region of the channel. Comparison with analytic solution.

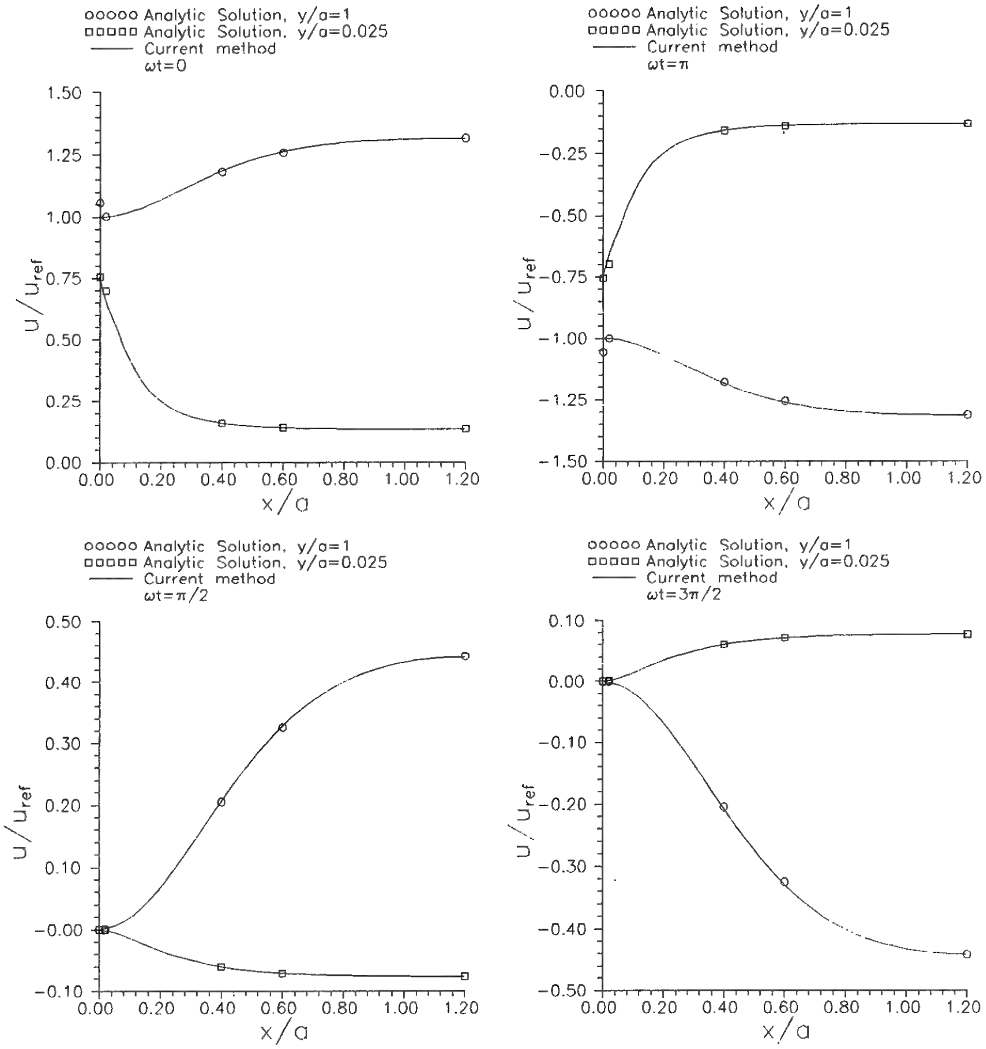


Figure 4. Longitudinal velocity component along the circular tube for one cycle of the flow.

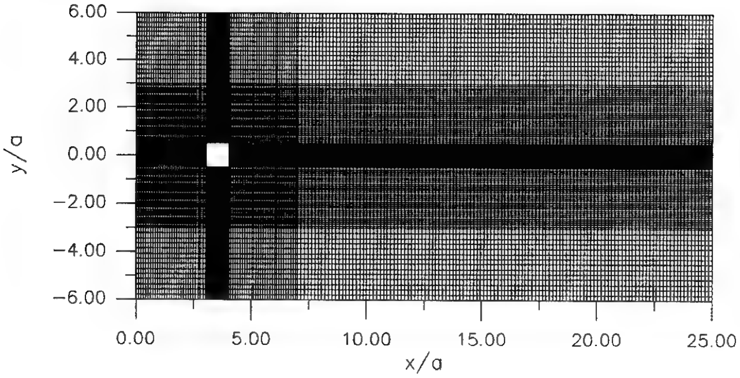


Figure 5. The 200×110 grid for the flow around a square cylinder.

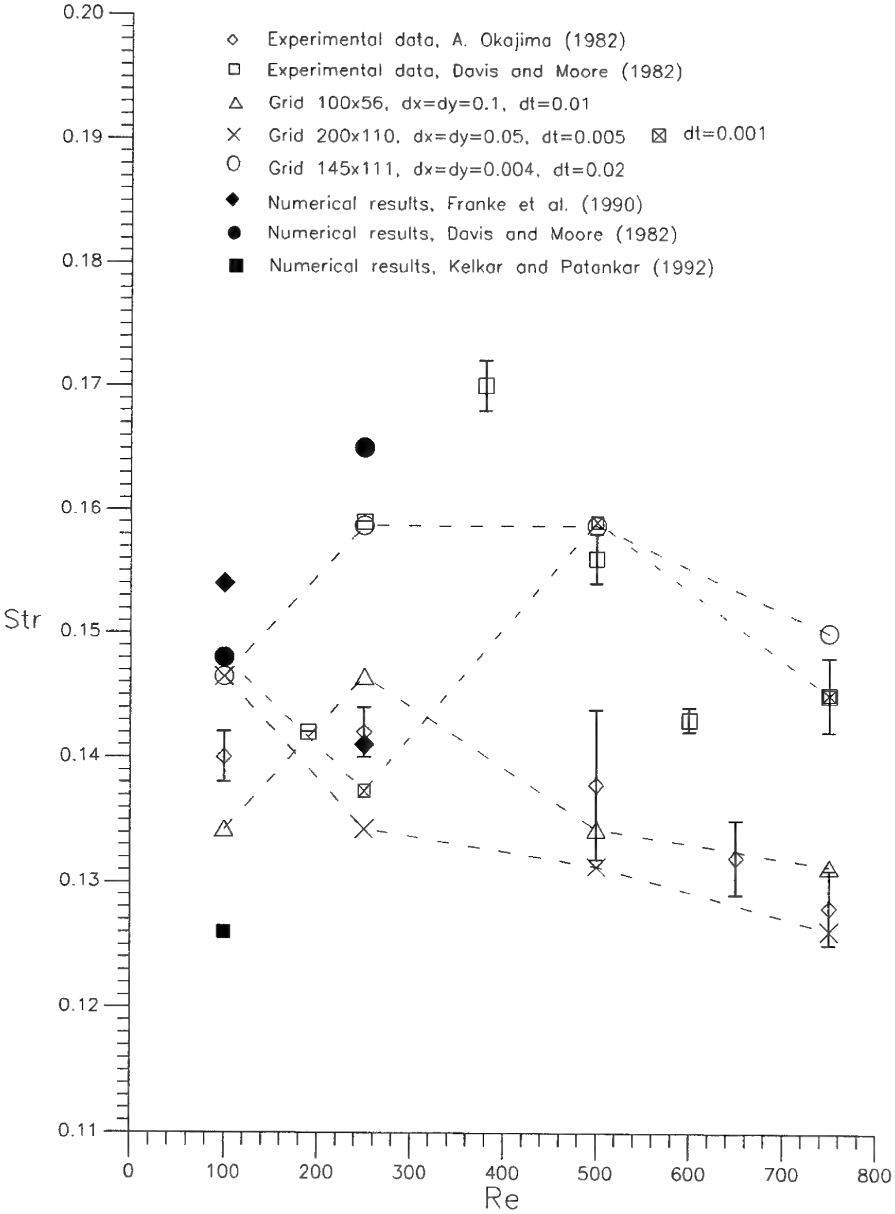


Figure 6. Strouhal number as function of the Reynolds number. Comparison with experimental data and numerical results.

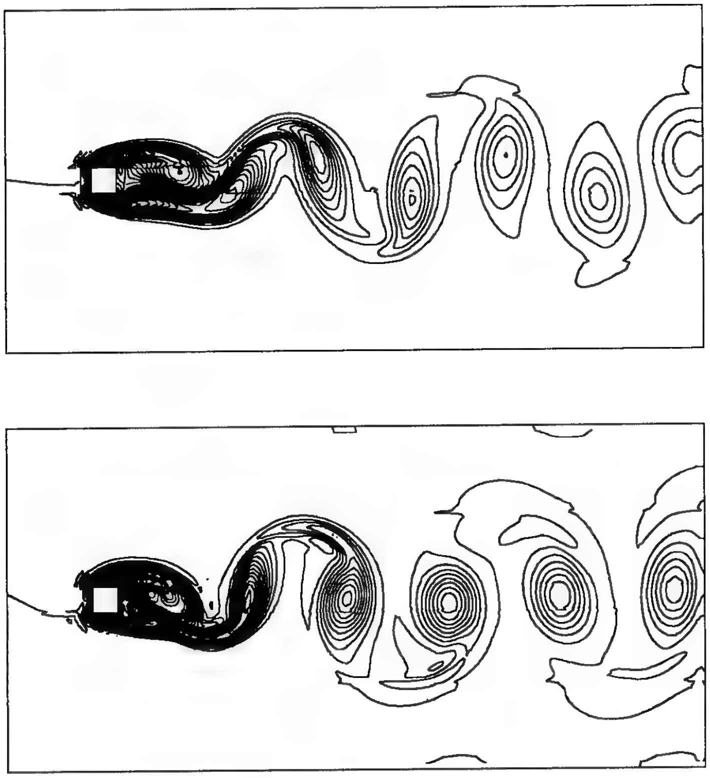


Figure 7. Vorticity isolines for Reynolds numbers 100 (up) and 250 (down). Grid 200×110.

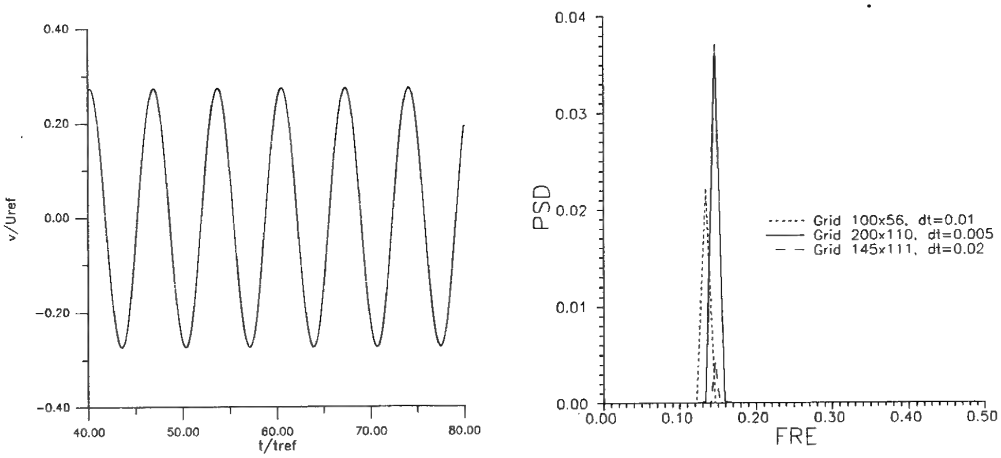


Figure 8. Time history of the v component of the velocity behind the square cylinder(left) and the corresponding power spectrum analysis (right).

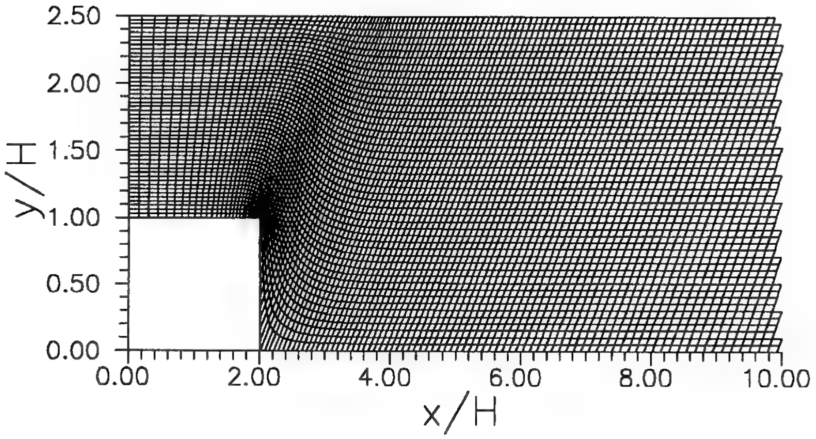
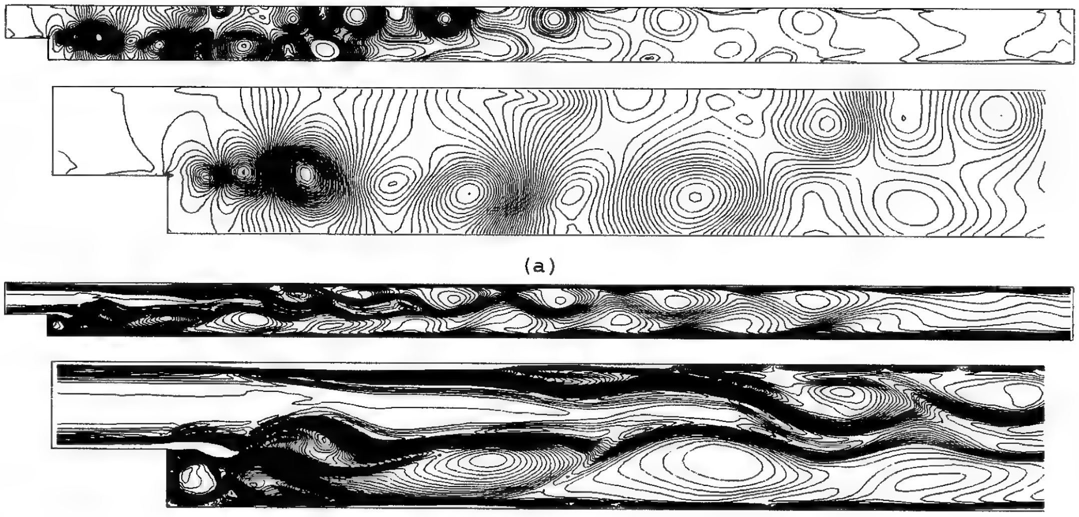


Figure 9. A view of the 250×50 grid used for the solution of the unsteady turbulent flow in the backward facing step.



(a)

(b)

Figure 10. Unsteady flow in a backward facing step. (a) pressure isolines, (b) vorticity isolines.

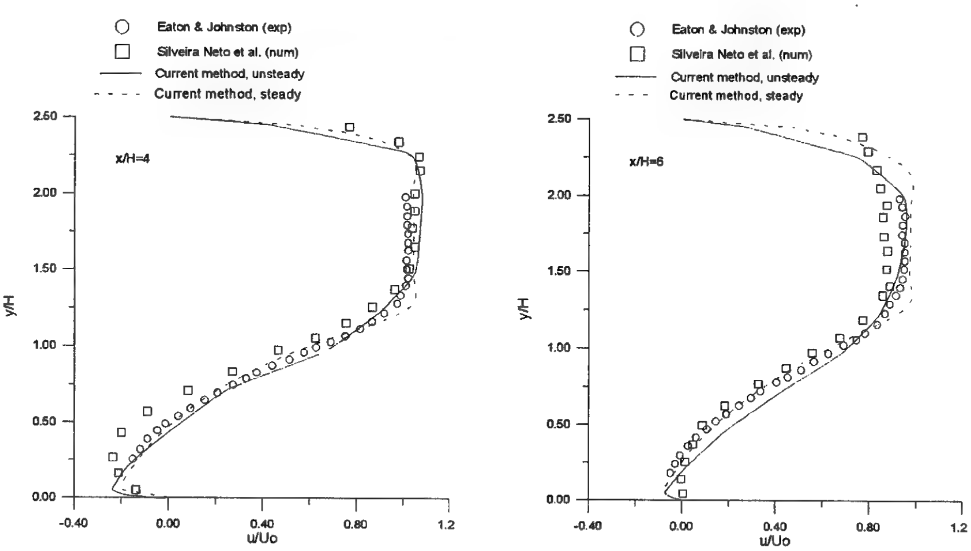


Figure 11. Time mean longitudinal velocity profiles versus experimental data and other numerical results.

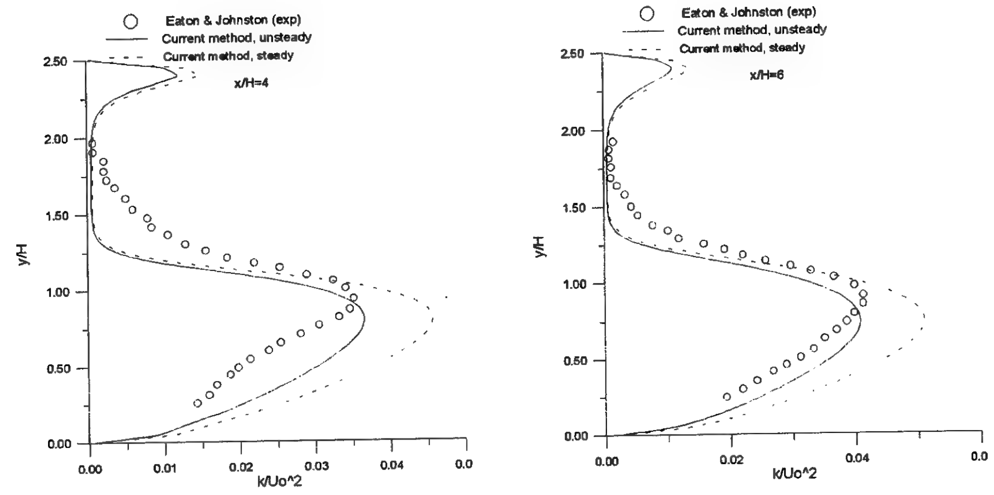


Figure 12. Time mean kinetic energy profiles versus experimental data.

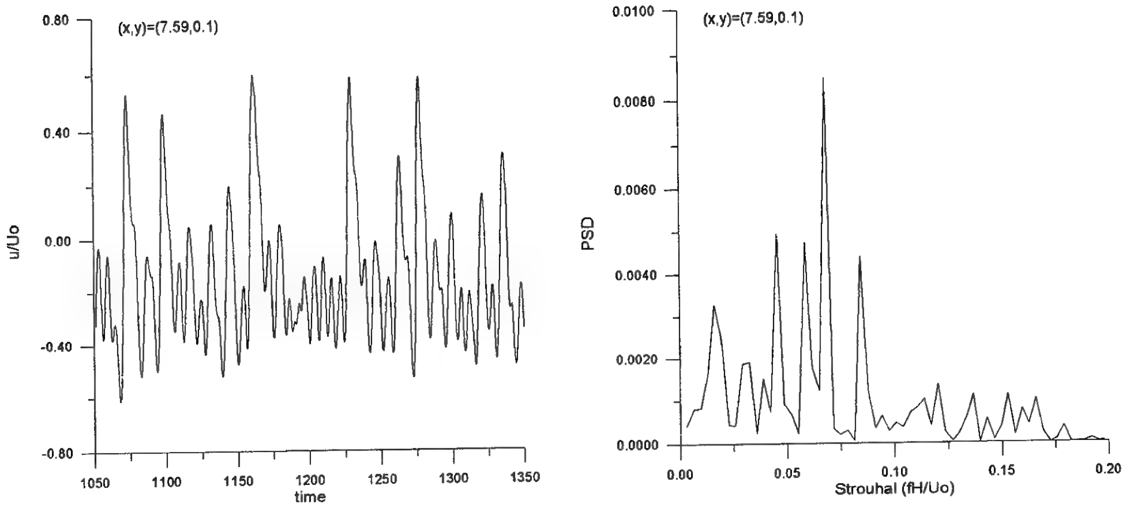


Figure 13. Time evolution of the longitudinal velocity near the lower wall (left) and the corresponding spectrum analysis (right).

ADAPTION BY GRID MOTION FOR UNSTEADY EULER AEROFOIL FLOWS

C. B. Allen

Department of Aerospace Engineering,
University of Bristol,
Bristol, BS8 1TR,
United Kingdom.

Abstract

A solution-adaptive structured grid technique is described for the computation of steady and unsteady Euler flows past aerofoils. Transfinite interpolation is used to generate the grids as this is well-suited to unsteady flows, since grid speeds required in the flux terms are available directly from the algebraic mapping. A novel approach to grid adaption is described. Adaption is performed by adapting the interpolation parameters, instead of the physical grid positions, so the adapted grid positions are available algebraically. Hence, the grid speeds required for unsteady computations are also available algebraically. For unsteady flows grid adaption is performed by imposing an 'adaption velocity' on grid points, thereby applying the adaption gradually over several time steps and avoiding the interpolation of the solution from one grid to another, associated with instantaneous adaption. Steady and unsteady aerofoil flows are considered. In both cases the adaptive grid technique is shown to produce sharper shock resolution for a very small increase in CPU requirements.

1 INTRODUCTION

Increases in computer power have meant that computational methods for unsteady flows have become commonplace. However, the CPU requirements of these methods can still be large. Moving grids are often used, and so repeated grid generation is required, and a large numerical integration time may be necessary to reach a periodic solution. Grid adaptivity is therefore desirable to improve solution resolution, in regions of high flow gradients, without significantly increasing the CPU requirements. There has been much recent discussion about whether structured or unstructured grids are best. Unstructured grids appear to have the advantage of lending themselves more naturally to grid adaption or enrichment, but the computational cost can be large, due to the grid connectivity data required. It has been shown [1] that for steady computations a solution computed using an unstructured grid requires 2 to 5 times the CPU time of that on a structured grid with the same number of nodes. The situation is likely to be worse for unsteady computations, where the grid must be recomputed at least once per time step.

This paper describes a solution-adaptive grid technique for steady and unsteady Euler flows using structured grids computed by the transfinite interpolation technique. Transfinite interpolation is well-suited to unsteady computations [2] since the grid speeds are available directly from the interpolation equation. The grid generation is remarkably simple, grid positions are obtained by interpolation of boundary positions and grid speeds by interpolation of boundary speeds, the interpolation being the same in each case.

Structured grid adaption is often achieved by solving a set of partial differential equations for the complete domain, for example Catherall [3] solves a combination of Laplace, Poisson, and equidistribution equations with source terms added to control grid stretching, spacing, and orthogonality. Pericleous *et al* [4] solve an equidistribution equation, based on solution gradients, along each grid line in each coordinate direction, then solve a Laplace equation for the resulting grid positions to ensure orthogonality. Although these approaches are suitable for steady flows, they are less suitable for unsteady flows. When considering moving grids the grid speeds are required in the flux evaluation, and neither approach leads to an obvious method of evaluating these speeds.

A different approach is presented here, wherein a new interpolation technique is developed and grid adaption is performed by adapting the interpolation parameters instead of the physical grid positions. In this way it is possible to determine the adapted grid positions algebraically. This represents a significant advantage when considering unsteady computations on moving grids. The resulting grid position equation can simply be differentiated with respect to time to yield the grid speeds algebraically.

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

However, there is a further problem encountered when adapting the grid during an unsteady computation. The conventional steady technique is to adapt the grid instantaneously and interpolate the solution to the new grid. This is less suitable for unsteady flows since many adaptions are required over several periods of motion, and repeated interpolation may result in a gradual loss of accuracy. Unstructured adaptive grids have been developed for unsteady flows, see for example [5, 6], and regions of high gradients are simply enriched with extra points. However, an interpolation step is still required, and this has been shown to lead to a conservation loss, even for unstructured grids, [5].

The adaption for unsteady flows is carried out here by imposing an 'adaption velocity' onto each grid point, thereby moving the grid points from one adapted grid position to the next over several time steps. This avoids the instantaneous adaption approach and so interpolation is not required. It also requires no extra grid generation.

2 UPWIND DIFFERENCE SCHEME

A finite-volume upwind scheme is used to solve the two-dimensional unsteady Euler equations in integral form, for the domain Ω with boundary $\partial\Omega$

$$\frac{\partial}{\partial t}\int\int_{\Omega} \mathbf{U} dx dy + \int_{\partial\Omega} (\mathbf{F} dy - \mathbf{G} dx) = 0. \qquad (1)$$

The vector of conserved variables $\mathbf{U}$ and convective fluxes $\mathbf{F}$ and $\mathbf{G}$, for moving grids, are;

$$\begin{aligned} \mathbf{U} &= [\rho, \rho u, \rho v, E]^T, && (2)\\ \mathbf{F} &= [\rho U, \rho u U + P, \rho v U, (E+P)U + x_t P]^T, && (3)\\ \mathbf{G} &= [\rho V, \rho u V, \rho v V + P, (E+P)V + y_t P]^T, && (4) \end{aligned}$$

and

$$U = u - x_t, \qquad V = v - y_t \qquad (5)$$

where x_t and y_t are the inertial grid speeds in the x and y directions respectively.

The cartesian velocity components normal and tangential to each computational cell face, and the contravariant velocity normal to the cell face, are then

$$\overline{u} = u\frac{\Delta y}{\Delta s} - v\frac{\Delta x}{\Delta s}, \qquad \overline{v} = u\frac{\Delta x}{\Delta s} + v\frac{\Delta y}{\Delta s}, \qquad (6)$$

$$\overline{U} = (u - x_t)\frac{\Delta y}{\Delta s} - (v - y_t)\frac{\Delta x}{\Delta s}. \qquad (7)$$

Here Δx and Δy are the cell face components and Δs is the face length. The general flux function in the direction normal to the cell face is then

$$\overline{\mathbf{F}} = [\rho\overline{U}, \rho\overline{U}\overline{u} + P, \rho\overline{U}\overline{v}, E\overline{U} + P\overline{u}]^T, \qquad (8)$$

and the flux across the face simply $\overline{\mathbf{F}}\Delta s$. This general flux vector is split into a forward part $\overline{\mathbf{F}}^+$ associated with positive moving waves only, i.e. all eigenvalues of $\frac{\partial\overline{\mathbf{F}}^+}{\partial\mathbf{U}} \geq 0$, and a backward part $\overline{\mathbf{F}}^-$ associated with negative moving waves only, all eigenvalues of $\frac{\partial\overline{\mathbf{F}}^-}{\partial\mathbf{U}} < 0$. At each cell face a pair of states are thus defined and a single numerical flux derived from this pair. The split flux components are, see Van-Leer [7] and Parpia [8],

$$\overline{\mathbf{F}}^{\pm} = \left\{ \begin{array}{l} f^{\pm}_{mass} \\ f^{\pm}_{mass}\cdot\left[\frac{(-\overline{U}\pm 2a)}{\gamma} + \overline{u}\right] \\ f^{\pm}_{mass}.\overline{v} \\ f^{\pm}_{energy} \end{array} \right\} \qquad (9)$$

where

$$f^{\pm}_{mass} = \pm\frac{\rho a}{4}(\overline{M} \pm 1)^2 \qquad (10)$$

$$f^{\pm}_{energy} = f^{\pm}_{mass}\left\{\frac{[(\gamma-1)\overline{U} \pm 2a]^2}{2(\gamma^2-1)} - \frac{\overline{U}^2}{2} + \frac{\overline{u}^2 + \overline{v}^2}{2} \right.$$
$$\left. + \left(x_t\frac{\Delta y}{\Delta s} - y_t\frac{\Delta x}{\Delta s}\right)\frac{(-\overline{U} \pm 2a)}{\gamma}\right\} \qquad (11)$$

and $\overline{M}$ the Mach number normal to the cell face $= \frac{\overline{U}}{a}$, and a is the local acoustic speed. The above splitting is only valid for $|\overline{M}| \leq 1$. Else

$$\overline{\mathbf{F}}^{+} = \overline{\mathbf{F}}, \quad \overline{\mathbf{F}}^{-} = 0, \quad if \ \overline{M} > 1, \qquad (12)$$

$$\overline{\mathbf{F}}^{+} = 0, \quad \overline{\mathbf{F}}^{-} = \overline{\mathbf{F}}, \quad if \ \overline{M} < -1. \qquad (13)$$

The general flux vector is split by

$$\overline{\mathbf{F}} = \overline{\mathbf{F}}^{+}(\mathbf{U}^+) + \overline{\mathbf{F}}^{-}(\mathbf{U}^-). \qquad (14)$$

A third-order spatial interpolation is used to evaluate $\mathbf{U}^+$ and $\mathbf{U}^-$ at each cell face, along with the continuously differentiable flux limiter due to Anderson *et al* [9].

Once $\overline{\mathbf{F}}$ has been split into its components the resulting flux must be rotated back to our original coordinate system. This is achieved by

$$\mathbf{F}\Delta y - \mathbf{G}\Delta x = R^{-1}[\overline{\mathbf{F}}^{+}(\mathbf{U}^+) + \overline{\mathbf{F}}^{-}(\mathbf{U}^-)]\Delta s \qquad (15)$$

where R is the rotation matrix.

An explicit three-stage Runge-Kutta scheme is used to integrate the equations forward in time. Local time-stepping is used for steady flows.

3 GRID GENERATION

Unsteady flows using structured moving grids will be considered. As the grid positions and speeds must be repeatedly calculated during an unsteady computation, we require a method of grid generation which is simple, and which gives the speeds algebraically rather than having to evaluate numerical differences between grid positions on successive time levels. It was thus decided to use the transfinite interpolation method originally described by Gordon and Hall [10]. For the vector function

$$\mathbf{f}(\eta,\xi) = [\mathbf{x}(\eta,\xi), \mathbf{y}(\eta,\xi)] \tag{16}$$

which is known only on certain lines of the region

$$\left.\begin{array}{ccccc} \eta_1 & \leq & \eta & \leq & \eta_2 \\ \xi_1 & \leq & \xi & \leq & \xi_2 \end{array}\right\} \tag{17}$$

transfinite interpolation gives the interpolated function $\mathbf{f}(\eta,\ \xi)$ throughout the region by a direct algebraic mapping. The general transfinite interpolation method results in a recursive algorithm, see Eriksson [11]. However, for a C-grid the inner and outer boundaries are lines of constant ξ where η is known. Defining one normal derivative only at the inner boundary, the algorithm reduces to ξ direction interpolation only,

$$\mathbf{f}(\eta,\xi) = \psi^0(\xi)\mathbf{f}(\eta,\xi_1)+\psi^1(\xi)\frac{\partial}{\partial\xi}\mathbf{f}(\eta,\xi_1)+\psi^2(\xi)\mathbf{f}(\eta,\xi_2) \tag{18}$$

Here $\psi^{0,1,2}$ are the blending functions in the ξ direction. The function $\mathbf{f}$ actually represents a transformation from (η,ξ) space to (x,y) space. The grid points are indexed by i and j in the η and ξ directions respectively, and then each i and j line are defined as constant η and ξ lines respectively. The variables are normalised such that

$$0 \leq \eta, \xi, \psi^{0,1,2} \leq 1. \tag{19}$$

The boundaries $\mathbf{f}(\eta, 0)$ and $\mathbf{f}(\eta, 1)$ are known at $imax$ discrete points, i.e. $\mathbf{f}_i(0)$ and $\mathbf{f}_i(1)$. The value of ξ at each constant ξ line is then defined as $j/jmax$. The blending functions ψ^0 and ψ^2 control the spacing in the ξ direction, and ψ^1 controls how far the normal direction affects the line direction. The most effective blending functions have been found to be

$$\xi = \frac{j}{jmax}, \tag{20}$$

$$\overline{\xi} = \left\{\frac{e^{\xi}-1-\xi}{e-2}\right\}^{st}, \tag{21}$$

$$\psi_j^0 = 1-\overline{\xi}, \tag{22}$$

$$\psi_j^1 = \sqrt{\overline{\xi}} - \left\{\frac{e^{\overline{\xi}}-1-\overline{\xi}}{e-2}\right\}, \tag{23}$$

$$\psi_j^2 = \overline{\xi}, \tag{24}$$

where st is a stretching exponent. The $imax \times jmax$ grid positions then come from

$$\mathbf{f}_{i,j} = \psi_j^0\mathbf{f}_i(0) + \psi_j^1\frac{\partial}{\partial\xi}\mathbf{f}_i(0) + \psi_j^2\mathbf{f}_i(1). \tag{25}$$

Figure 1(a) shows the grid near a NACA0012 aerofoil, resulting from the above interpolation, using $imax = 129$, (99 points on the aerofoil surface, 15 in the wake either side), $jmax = 30$, $st = 1.2$, and the outer boundary is 20 chords away. Figure 1(b) shows the corresponding variation of η_i and ψ_j^2. (Grid points $i = 13 - 117, j = 1 - 13$ are shown).

By differentiating (25) with respect to time the grid speeds can be obtained analytically, (blending functions assumed constant, and outer boundary fixed)

$$\frac{d}{dt}\mathbf{f}_{i,j} = \psi_j^0\frac{d}{dt}\mathbf{f}_i(0) + \psi_j^1\frac{d}{dt}\left\{\frac{\partial}{\partial\xi}\mathbf{f}_i(0)\right\}. \tag{26}$$

Hence, grid positions are calculated by interpolation of the boundary positions, and grid speeds by interpolation of boundary speeds, the interpolation being the same in each case.

4 GRID ADAPTION

The grid is to be adapted, according to the solution, so that grid points are clustered in regions of high gradients. Adaption is normally performed in (x,y) space. However, while this gives suitable grids for steady computations, the grid positions, and hence more importantly grid speeds, would not be available algebraically for unsteady computations. Only numerical values of dx/dt and dy/dt could be evaluated between different adapted grids during an unsteady computation, and these could cause problems of grid distortion and crossover when grid points move along highly curved lines.

Adaption is achieved here by writing the interpolation function in a more general form and adapting the interpolation parameters instead of the physical coordinates, such that grid positions are available algebraically.

Since each i line is a constant η line, we can move points along an i line by simply varying ξ (or ψ) along that line. The line remains unchanged, only the distribution of points along it is altered. Adaption in the ξ direction is thus achieved by letting the blending functions be variant in η as well as ξ, and so we have,

$$\mathbf{f}_{i,j} = \psi_{i,j}^0\mathbf{f}_i(0) + \psi_{i,j}^1\frac{\partial}{\partial\xi}\mathbf{f}_i(0) + \psi_{i,j}^2\mathbf{f}_i(1). \tag{27}$$

Figure 2(a) shows the near aerofoil grid resulting from varying st from 1.1 to 1.3 depending on the η spacing, i.e. clustering points near the leading and trailing edges, and figure 2(b) the corresponding $\eta_i, \psi_{i,j}^2$ variation. To adapt in the other direction, we must now change the interpolation so that each i line is no longer constrained to be a line of constant η. Along each j line η is now varied to give the required distribution in this direction (previously η_i was the same on every j line). The inner and outer boundaries, $\mathbf{f}(\eta, 0)$ and $\mathbf{f}(\eta, 1)$ are determined in terms of η, so that they are known at any point, not just the specified points $\mathbf{f}_i(0)$. The interpolation is then

$$\mathbf{f}_{i,j} = \psi^0_{i,j}\mathbf{f}(\eta_{i,j}, 0) + \psi^1_{i,j}\frac{\partial}{\partial\xi}\mathbf{f}_i(0) + \psi^2_{i,j}\mathbf{f}(\eta_{i,j}, 1). \quad (28)$$

By adapting η and ψ^2 instead of x and y the grid positions are still available algebraically. This means that the grid speeds are also available algebraically, which is essential for efficient unsteady adaption.

4.1 Adaption in Each Direction

Instead of computing a completely new grid due to adaption, it is desirable to simply change only a small region of the grid where adaption is required.

Adaption in the j direction is achieved by varying ψ^2 along each i line. For adaption in the i direction η is changed along each j line to give the required distribution.

Adaption is required in regions where flow quantity gradients are high, and the local Mach number gradient is used as a sensor. At each point the Mach number gradient in each direction is evaluated,

$$\left|\frac{\partial M}{\partial s_\eta}\right| = \left|\frac{M_{i,j} - M_{i-1,j}}{\Delta s_\eta}\right| \quad (29)$$

$$\left|\frac{\partial M}{\partial s_\xi}\right| = \left|\frac{M_{i,j} - M_{i,j-1}}{\Delta s_\xi}\right| \quad (30)$$

where

$$\Delta s_\eta = \sqrt{(x_{i,j} - x_{i-1,j})^2 + (y_{i,j} - y_{i-1,j})^2} \quad (31)$$

$$\Delta s_\xi = \sqrt{(x_{i,j} - x_{i,j-1})^2 + (y_{i,j} - y_{i,j-1})^2}. \quad (32)$$

The gradients at each point are normalised by the largest value over the domain. If this gradient is greater than a threshold value then adaption is deemed to be required at that point. There will usually be regions of points where adaption is required, i.e. 2/3 points around a shock and 5 to 10 points around a stagnation point, and so in each region the point with the largest Mach number gradient is identified. At each adaption point the spacing of grid points is controlled by defining two spacing factors, fr_1 for stagnation points, and fr_2 for other adaption points. Since parallel lines in (η, ψ^2) space may not be parallel in (x, y) space, the spacing at adaption points must be scaled thus,

$$\Delta\psi^2 = \frac{\Delta\psi^2_0}{fr_1}\frac{\Delta s_{\xi_0}}{\Delta s_\xi} \quad or \quad \Delta\psi^2 = \frac{\Delta\psi^2_0}{fr_2}\frac{\Delta s_{\xi_0}}{\Delta s_\xi}, \quad (33)$$

$$\Delta\eta = \frac{\Delta\eta_0}{fr_1}\frac{\Delta s_{\eta_0}}{\Delta s_\eta} \quad or \quad \Delta\eta = \frac{\Delta\eta_0}{fr_2}\frac{\Delta s_{\eta_0}}{\Delta s_\eta}, \quad (34)$$

where $2.0 \leq fr_1, fr_2 \leq 5.0$, and $\Delta\eta_0$, $\Delta\xi_0$, Δs_{η_0}, and Δs_{ξ_0} are the initial spacings.

Consider, for example, the variation of η along the aerofoil surface, $\psi^2 = 0$. An intermediate variable, ζ, is defined so that $\eta = \eta(\zeta)$ where clearly $0 \leq \zeta \leq 1$. A uniform distribution of ζ is used, and then $\eta(\zeta)$ is defined to give the required distribution of points. Figure 3(a), shows the initial distribution of η along the aerofoil for 99 points on the aerofoil. This is the unadapted distribution of points on the aerofoil.

For a solution where adaption is required in the η direction, if for example a normal shock is present, $\Delta\eta$ is defined at that point using equation (34) and then use a cosine variation in η to get back to the unadapted distribution of η in as few points as possible. Figure 3(b) shows the variation of η along the aerofoil surface for the flow considered in the next section, when normal shocks are present at approximately 0.64 chord on the upper surface and 0.32 chord on the lower. This simple sampling and adaption procedure is performed for each line in each direction.

5 STEADY FLOW RESULTS

The steady flow over a NACA0012 aerofoil at 1.25° incidence, in a flow of freestream Mach number 0.8 is considered. The initial grid is similar to that shown in Figure 2, i.e. 129 × 30 C-grid, with *st* varying between 1.1 and 1.3 (clustering near the leading and trailing edges).

Figure 4(a) shows the pressure coefficient over the aerofoil computed on the non-adaptive grid, the dashed line is the reference AGARD solution [12].

The grid was then adapted by applying the Mach number gradient check along each line (in x, y space) and simply clustering points (in η, ψ^2 space) where this is greater than the threshold level. Figure 5 shows the resulting near-aerofoil variation of η and ψ^2, and the corresponding grid. The variation in the ψ^2 direction is unchanged, since where the Mach number gradient is above the threshold level the grid spacing is at the minimum value already. Clearly the grid needs to be smoothed. This is often done by solving a Laplace equation for the grid point coordinates (see for example [4]). However, that is not required here. As the grid was initially smooth a smoothing can be applied to the whole grid in each direction and the unadapted regions of the grid will be unaffected. A simple three-point smoothing is applied

$$\eta_{i,j} = f\left(\eta_{i,j-1}, \eta_{i,j}, \eta_{i,j+1}\right) \quad (35)$$

$$\psi^2_{i,j} = f\left(\psi^2_{i-1,j}, \psi^2_{i,j}, \psi^2_{i+1,j}\right). \quad (36)$$

Figure 6 shows the smoothed variation of η and ψ^2, and the corresponding grid. Figure 4(b) shows the surface pressure coefficient computed on the adapted grid. The improved shock capturing is clear.

In many steady flow adaption procedures the grid is allowed to adapt gradually by effectively progressing with the solution, until a time-asymptotic grid and solution are reached. The grid adaption here is only applied once, as this will be the case when an unsteady solution is periodically sampled as in section 7. The adaption only has one 'chance' to compute a suitable grid at each adaption point.

6 UNSTEADY EULER METHOD

The explicit time-stepping scheme used for steady flows can be made time-accurate by using a global time-step, and applied to unsteady motion on a moving mesh by incorporating the cell area changes at each stage in the time-stepping scheme [13]. However for a typical unsteady computation, with the grid size above, as many as 15000 time-steps, and two CPU hours, per period may be required. It is more efficient to solve the unsteady problem as a series of pseudo-steady problems. The implicit form of the differential equation for each computational cell is

$$\frac{\partial(A^{n+1}\mathbf{U}^{n+1})}{\partial t} + \mathbf{R}(\mathbf{U}^{n+1}) = 0 \qquad (37)$$

where A is the cell area and $\mathbf{R}$ is the upwinded flux integral. The implicit temporal derivative is then approximated by a second-order backward difference, following Jameson [14], giving

$$\frac{3}{2\Delta t}\left[A^{n+1}\mathbf{U}^{n+1}\right] - \frac{2}{\Delta t}\left[A^n\mathbf{U}^n\right] + \frac{1}{2\Delta t}\left[A^{n-1}\mathbf{U}^{n-1}\right] + \mathbf{R}(\mathbf{U}^{n+1}) = 0. \qquad (38)$$

A new residual $\mathbf{R}^*(\mathbf{U})$ is defined as

$$\mathbf{R}^*(\mathbf{U}) = \frac{3}{2\Delta t}\left[A^{n+1}\mathbf{U}\right] - \frac{2}{\Delta t}\left[A^n\mathbf{U}^n\right] + \frac{1}{2\Delta t}\left[A^{n-1}\mathbf{U}^{n-1}\right] + \mathbf{R}(\mathbf{U}) \qquad (39)$$

and then a new differential equation can be written in terms of a fictitious time τ,

$$A^{n+1}\frac{d\mathbf{U}}{d\tau} + \mathbf{R}^*(\mathbf{U}) = 0. \qquad (40)$$

This is simply time-marched to convergence in the fictitious time τ, for each real time-step. There is now no limit to the size of the real time step, Δt, that can be taken and this leads to a large reduction in CPU times. The time step is now limited by accuracy rather than stability. For each real time step equations (40) are solved to convergence using an implicit form of the three stage time-stepping scheme with local time-stepping that is used for steady computations. This approach also means that the grid generation routine only needs to be called once every real time-step, to calculate the grid positions and speeds at the next time level.

6.1 Consideration of Cell Area Changes

If the cell areas at each time level or stage are simply calculated using the instantaneous physical coordinates of the cell faces a numerical error is introduced which will increase with time. The cell areas must therefore satisfy a geometric conservation law of the same integral form as the mass conservation law [15], i.e.

$$\frac{\partial}{\partial t}\int\int_{\Omega} dx\,dy = \int_{\partial\Omega}(x_t dy - y_t dx) \qquad (41)$$

and this must be solved using the same numerical scheme as for the flow quantities. The cell areas at the next real time level are thus calculated by

$$A^{n+1} = \frac{4A^n}{3} - \frac{A^{n-1}}{3} + \frac{2\Delta t}{3}\sum_{k=1}^{4}\{x_{t_k}\Delta y_k - y_{t_k}\Delta x_k\}^{n+1} \qquad (42)$$

where $k = 1, 2, 3, 4$ represents the four cell faces.

7 UNSTEADY GRID ADAPTION

The normal steady flow adaption procedure is to compute the solution, sample it, and change the grid instantaneously. However, whether using structured grids, where a fixed number of grid points are redistributed to be clustered in regions of high gradient, or unstructured grids, where extra points are simply added in regions of high gradient, adaption results in grid points where the solution is not known. This then requires the interpolation of the solution from the old grid to the new. The repeated adaption and interpolation required over several periods in an unsteady computation can result in a gradual degeneration of the solution [5].

Also, the implicit scheme implemented here uses values of conserved variables and cell areas from previous time levels, which do not exist once the grid has been adapted, so instantaneous adaption cannot be applied to the unsteady solver used here.

To avoid this the grid adaption is spread over several (real) time steps and the motion of each point described in terms of an 'adaption velocity'. The periodic nature of the unsteady solution is exploited by sampling the solution over one period and adapting the grid accordingly over the next period. Therefore over one unsteady period the solution is sampled every *nsamp* real time steps, and the resulting adapted (η, ψ^2) distribution stored. When calculating the solution on the next period, over each set of *nsamp* time steps the velocity of each point required for that point to reach its position at the next adapted grid is imposed on each point, and the grid moves gradually between each adapted state.

If k is the adaption index ($k = 0, .., nadapt$, where $nadapt = nt/nsamp$ and nt is the number of real time steps per period), then $\eta_{i,j}^{(k)}, \psi_{i,j}^{2(k)}$ is the grid point distribution at adaption k. To move the grid points from one distribution to the next over $nsamp$ time steps we calculate the speed of each point, in (η, ψ^2) space

$$\frac{d\eta_{i,j}}{dt} = \frac{\eta_{i,j}^{(k+1)} - \eta_{i,j}^{(k)}}{nsamp\Delta t}, \qquad (43)$$

$$\frac{d\psi_{i,j}^2}{dt} = \frac{\psi_{i,j}^{2(k+1)} - \psi_{i,j}^{2(k)}}{nsamp\Delta t}. \qquad (44)$$

The grid speeds are obtained by differentiating equation 28 with respect to time,

$$\frac{d}{dt}\mathbf{f}_{i,j} = \frac{d\psi_{i,j}^0}{dt}\mathbf{f}(\eta_{i,j},0) + \frac{d\psi_{i,j}^1}{dt}\frac{\partial}{\partial\xi}\mathbf{f}_i(0) + \frac{d\psi_{i,j}^2}{dt}\mathbf{f}(\eta_{i,j},1)$$
$$+\psi_{i,j}^0\frac{d}{dt}\mathbf{f}(\eta_{i,j},0) + \psi_{i,j}^1\frac{d}{dt}\left\{\frac{\partial}{\partial\xi}\mathbf{f}_i(0)\right\} + \psi_{i,j}^2\frac{d}{dt}\mathbf{f}(\eta_{i,j},1). \qquad (45)$$

Then superimposing the adaption speeds onto the unsteady motion speeds, and replacing $\frac{d}{dt}$ by $\frac{d\eta}{dt}\frac{d}{d\eta}$ where required, we obtain (the outer boundary is fixed in time so $\frac{d}{dt}\mathbf{f}(\eta, 1) = 0$ due to motion)

$$\frac{d}{dt}\mathbf{f}_{i,j} = \frac{d\psi_{i,j}^0}{dt}\mathbf{f}(\eta_{i,j},0) + \frac{d\psi_{i,j}^1}{dt}\frac{\partial}{\partial\xi}\mathbf{f}_i(0) + \frac{d\psi_{i,j}^2}{dt}\mathbf{f}(\eta_{i,j},1)$$
$$+\psi_{i,j}^0\left\{\frac{d\eta_{i,j}}{dt}\frac{d}{d\eta}\mathbf{f}(\eta_{i,j},0) + \frac{d\eta_{i,j}}{dt}\frac{d}{d\eta}\mathbf{f}(\eta_{i,j},0)^M\right\}$$
$$+\psi_{i,j}^1\left\{\frac{d\eta_{i,j}}{dt}\frac{d}{d\eta}\left\{\frac{\partial}{\partial\xi}\mathbf{f}_i(0)\right\} + \frac{d}{dt}\left\{\frac{\partial}{\partial\xi}\mathbf{f}_i(0)^M\right\}\right\}$$
$$+ \psi_{i,j}^2\frac{d\eta_{i,j}}{dt}\frac{d}{d\eta}\mathbf{f}(\eta_{i,j},1) \qquad (46)$$

where the superscript M represents speeds due to the aerofoil motion. The implicit code is run with the unadapted grid for two periods, the adaptive grid data being stored during the $nadapt$ samples of the second period, and then two periods of adaptive grid computations are performed.

8 UNSTEADY RESULTS

The scheme was applied to the Mach 0.755 flow about a NACA0012 aerofoil pitching about quarter chord. The aerofoil motion is defined by

$$\alpha = 0.016^o + 2.51^o sin(\omega t) \qquad (47)$$

The reduced frequency parameter, $k = \frac{\omega c}{2U_\infty}$, was 0.0814 where c is the aerofoil chord, and U_∞ is the undisturbed flow speed. The scheme was run at a CFL number, based on τ of 1.4, and local time stepping was used to accelerate convergence within each real time step. There were 180 real time steps per period and the same grid data was used as previously, 129 x 30 points, with 99 points on the aerofoil. In the adaptive computation $nsamp$ was 10 and so $nadapt$ was 18.

Figure 7 shows normal force and moment (about $\frac{1}{4}$ chord) coefficient loops obtained by the implicit method, adaptive and non-adaptive, and from experiment [16]. The coefficient loops are quite similar, but the adaptive C_n loop is slightly narrower, and the C_m loop has larger 'steps', than the standard solution. The instantaneous pressure distributions are shown in figure 8. The improved shock capturing with the adaptive grid is clear. Figure 9 shows the near aerofoil adaptive grid at each of the incidences considered in figure 8.

The non-adaptive scheme required 19 CPU minutes per period on a Stardent 3000 machine, and the adaptive grid solution approximately 22 CPU minutes. An explicit time-stepping scheme required approximately 15000 time-steps and two CPU hours per period [13]. Thus the implicit method requires only one-fifth of the CPU time of an explicit scheme, even with an adaptive grid.

9 CONCLUSIONS

Steady and unsteady solutions have been computed using non-adaptive and adaptive grids generated by a new transfinite interpolation technique. Grid adaption is performed by adapting the interpolation parameters, instead of the physical grid positions, so that the adapted grid positions are still available algebraically. This interpolation has been shown to be ideal for generating structured moving grids, since it is very simple, thus requires little CPU time, and since the grid speeds, even for adapted grids, are available directly from the interpolation equations. The simplicity of the interpolation results in great flexibility, and we can adapt the grid during an unsteady computation by imposing an 'adaption velocity' onto each grid point, thus performing adaption gradually. This avoids the interpolation of the solution from the old grid to the new associated with instantaneous adaption.

An upwind Euler scheme is used to compute the solutions. This is implemented using a dual-time implicit method for unsteady flows which is very efficient, requiring only $\frac{1}{5}$ the CPU time of the explicit scheme.

For steady and unsteady aerofoil computations, the adaptive grid method produces sharper solutions for very little increase in CPU requirements.

Currently, only a fairly crude grid redistribution technique is employed. Future work will include developing a more sophisticated method, along with extending the adaptive technique into three dimensions. The method should be equally simple, the only difficulty arising from the third dimension being that the boundary definition will involve determining spline equations for surfaces rather than lines.

References

[1] Salas, M. D. (ed.), "Accuracy of Unstructured Grid Techniques Workshop", (NASA Langley Research Centre, Hampton, VA), Jan. 1990, (NASA Proceedings to be published)

[2] Williams, A. L. and Fiddes, S. P., "Solution of the 2-D Unsteady Euler Equations on a Structured Moving Grid", Bristol University Aero. Eng. Dept. Report 453, 1992.

[3] Catherall, D., "Adaptivity Through Mesh Movement", in Proceedings of European Forum, *Recent Developments and Applications in Aeronautical CFD,* Bristol, 1993.

[4] Patel, M. K., Pericleous, K. A. and Baldwin, S., "The Development of a Structured Mesh Grid Adaption Technique For Resolving Shock Discontinuities in Upwind Navier-Stokes Codes", Int J Numer Meth Fluids, Vol. 20, 1995.

[5] Morgan, K., "Unstructured Mesh Methods", in Proceedings 16, Rutherford Appleton Laboratory EASE Community Club in CFD *New Opportunities and Directions in Aeronautical CFD,* April 1994.

[6] Webster, B. E., Shephard, M. S., Rusak, Z., and Flaherty, J. E., "Unsteady Compressible Airfoil Aerodynamics Using an Adaptive Time-Discontinuous GLS Finite Element Method", AIAA Paper 93-0339.

[7] Van-Leer, B., "Flux-Vector Splitting for the Euler Equations", Lecture Notes in Physics, Vol. 170, 1982, pp. 507-512.

[8] Parpia, I. H., "Van-Leer Flux-Vector Splitting in Moving Coordinates", AIAA J, Vol. 26, January 1988, pp. 113-115.

[9] Anderson, W. K. Thomas, J. L. and Van-Leer, B., "Comparison of Finite Volume Flux Vector Splittings for the Euler Equations", AIAA J, Vol. 24, September 1986, pp. 1453-1460.

[10] Gordon, W. J. and Hall, C. A., "Construction of Curvilinear Coordinate Systems and Applications of Mesh Generation", Int J Numer Meth Eng, 1973, Vol. 7, pp461-477.

[11] Eriksson, L. E., "Generation of Boundary-Conforming Grids Around Wing-Body Configurations Using Transfinite Interpolation", AIAA J, Vol. 20, No. 10, 1982, pp. 1313-1320.

[12] (Anonymous) "Test Cases for Inviscid Flow Field Methods", AGARD Agardograph AR-211, 1985.

[13] Allen, C. B., "Central-Difference and Upwind-Biased Schemes for Steady and Unsteady Euler Aerofoil Computations", Aero. J, Vol. 99, 1995.

[14] Jameson, A., "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings", AIAA Paper 91-1596.

[15] Thomas, P. D. and Lombard, C. K., "Geometric Conservation Law and its Application to Flow Computations on Moving Grids", AIAA J, Vol. 17, October 1979, pp. 1030-1037.

[16] (Anonymous) "Compendium of Unsteady Aerodynamic Measurements", AGARD-R-702, 1982.

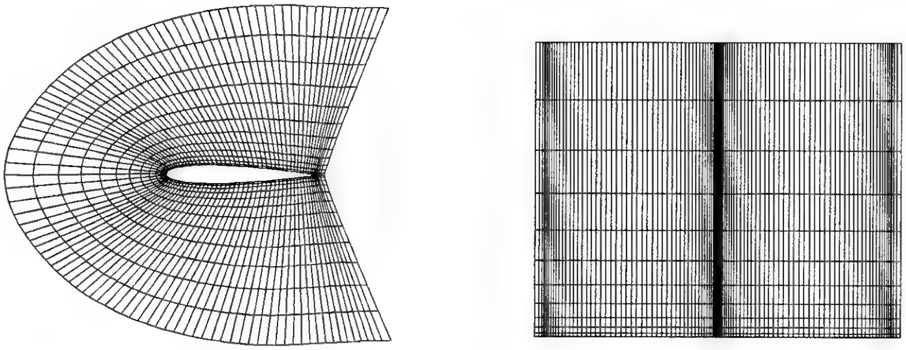


Fig.1. *Near Aerofoil Grid (a)* (x, y) *and (b)* (η, ψ^2).

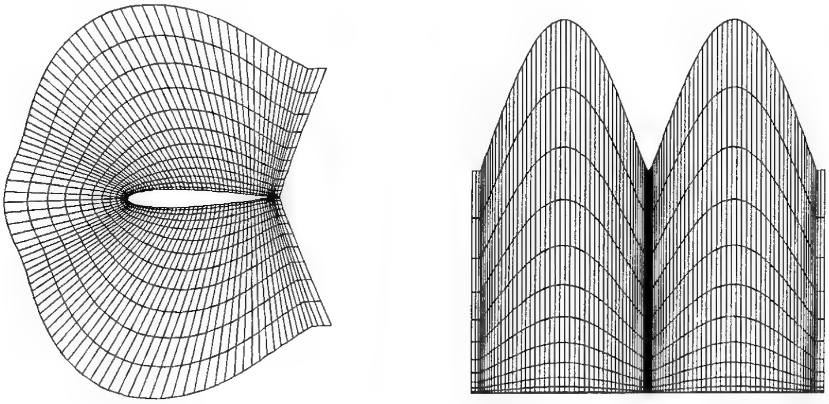


Fig.2. *Near Aerofoil Grid (a)* (x, y) *and (b)* (η, ψ^2).

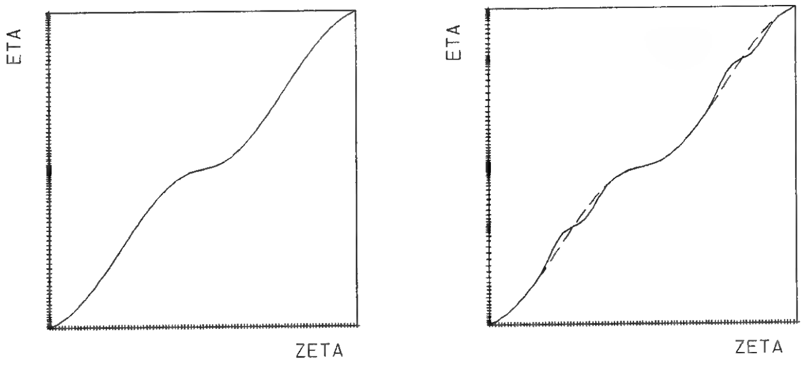


Fig.3.*(a) Unadapted, and (b) Adapted, Variation of* η.

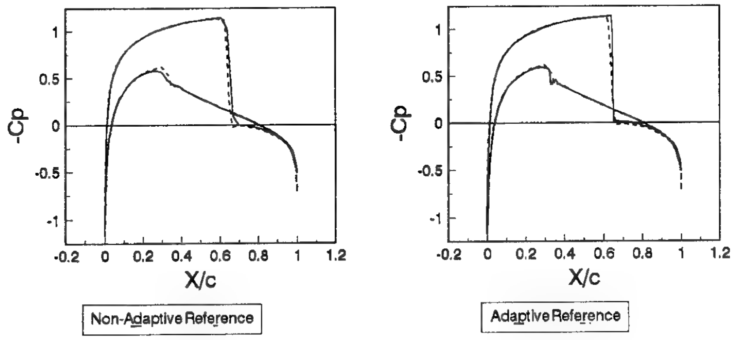


Fig.4. *Surface* C_p, *(a) Non-Adaptive and (b) Adaptive,* $NACA0012$, $M = 0.8, \alpha = 1.25^o$.

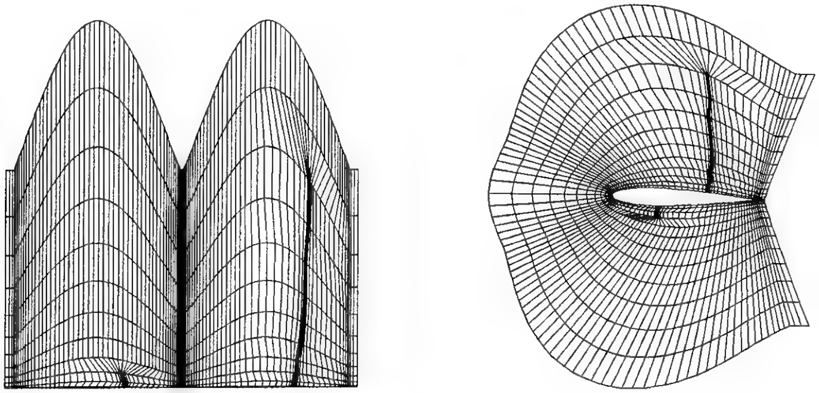


Fig.5. *Near Aerofoil Adapted Grid (a)* (η, ψ^2) *and (b)* (x, y).

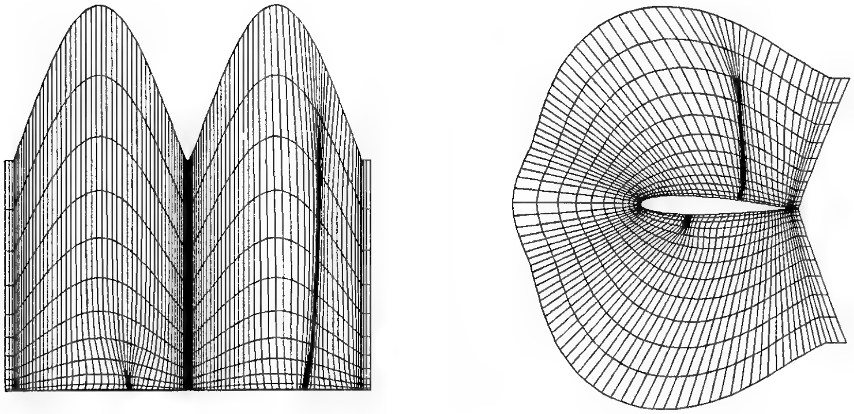


Fig.6. *Near Aerofoil Smoothed Adapted Grid (a) (η, ψ^2) and (b) (x, y).*

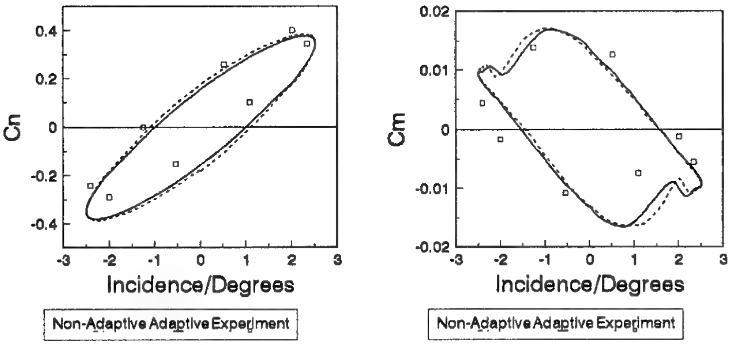


Fig.7. *Unsteady Normal Force and Moment Coefficient.*

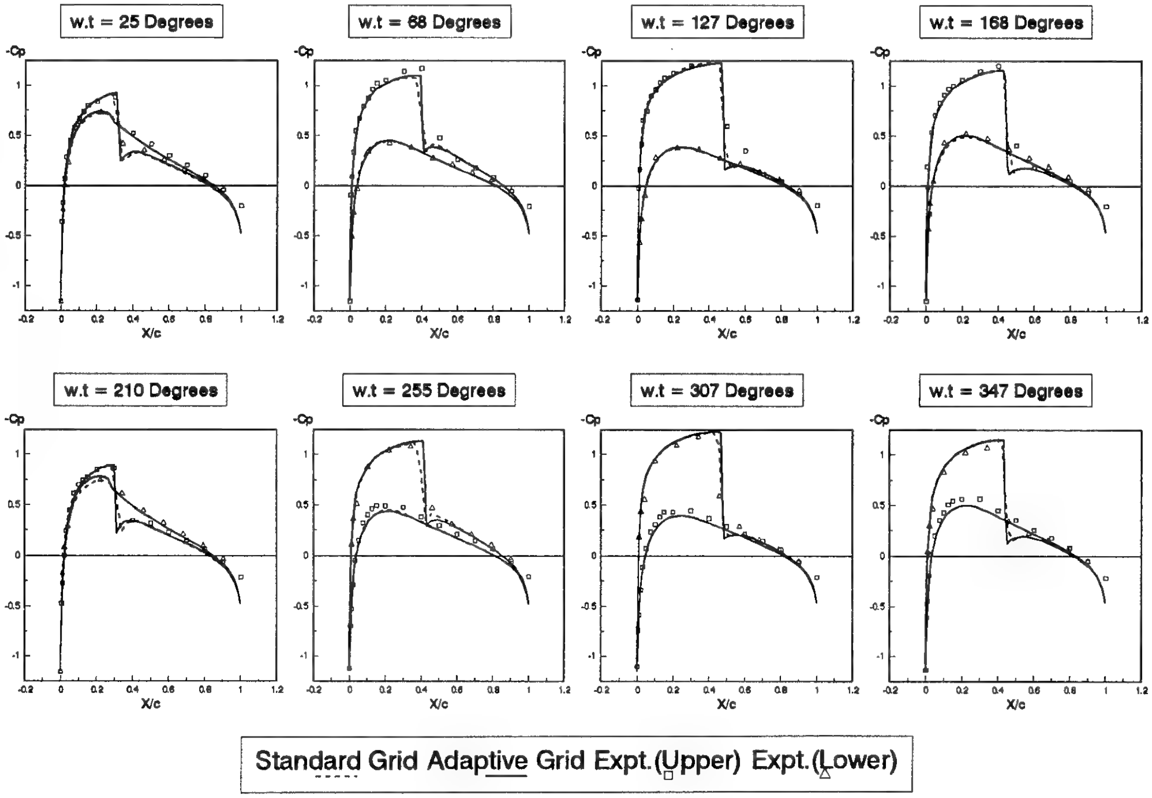


Fig.8. *Instantaneous Pressure Distributions.*

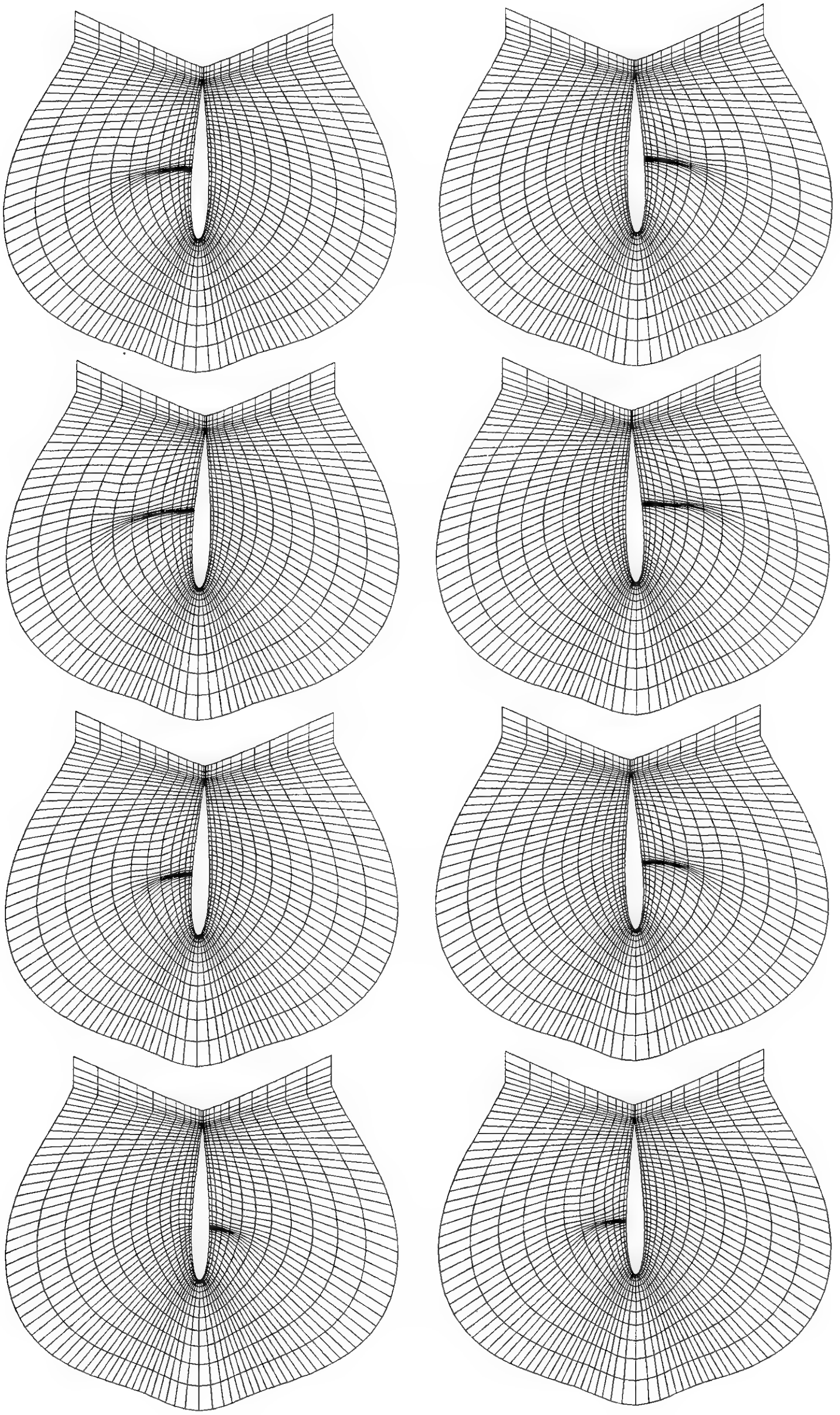


Fig.9. *Instantaneous Near Aerofoil Adaptive Grids.*

Adaptive Computation of Unsteady Flow Fields with the DLR-$\mathcal{T}$-Code

O. FRIEDRICH, D. HEMPEL, A. MEISTER, TH. SONAR
Deutsche Forschungsanstalt für Luft- und Raumfahrt e.V.
Institut für Strömungsmechanik
Bunsenstraße 10, D-37073 Göttingen
Fax: +49 551 709 2446

1 Abstract

The features and abilities of the DLR-τ-code, a finite volume approximation of box type for the Navier-Stokes equations governing viscous, compressible fluid flow, are described in detail. The code is able to compute flow in moving reference frames and is build upon dynamically adaptive concepts to allow for grid refinement in the framework of non-stationary aerodynamics. Implicit as well as explicit time-stepping schemes can be used depending on the kind of application.

2 Introduction

The DLR-τ-Code is a finite volume approximation of the Navier-Stokes equations governing compressible, viscous flow. The method uses a box-type discretisation and works on general conforming triangulations. The discretisation of the convective fluxes is accomplished by means of an approximate Riemann solver while the diffusive fluxes are discretised in a central manner.

To achieve high resolution recovery techniques of ENO-type are applied. New recovery techniques are presented which are based on radial basis functions. Although their use is restricted by now to small problems, it can be shown that they obey a certain optimality condition.

Explicit time stepping through TVD-Runge-Kutta methods is used in a parallelized version of the code. This parallelized version includes an intelligent load balancer for performance-controlled domain decomposition and can handle arbitrary message passing libraries like PVM or P4.

To effectively deal with unsteady flow problems such as pitching airfoils and moving bodies in general, the implementation of implicit time stepping schemes is also.considered. The development of an implicit method on unstructured grids leads to an linear system of equations with a large sparse and badly conditioned matrix. In this case, the fundamental mathematical assignment is the discreption of a fast solver for such linear systems of equations. Extensive investigations with several possible algorithms indicated the superiority of a pre-conditioned GMRES algorithm. The preconditioner is a simple incomplete LU-factorization which dramatically improves the convergence properties of GMRES in the case of the Euler equations. Experience gained by numerical investigations has shown that even fast unsteady flow phenomena like moving shocks in channels can be effectively treated by this combination of algorithms.

The τ-Code employes dynamically adaptive strategies based on insertion and removing of grid points. Conservative interpolation avoids mass errors during the process of adaptation. One of the main design goals was the use of reliable error indicators instead of refinement indicators based on gradients of flow variables. The indicators we consider are based on the finite element residual of the Euler equations.

3 Governing equations

We consider the Navier-Stokes equations in a moving reference frame. In this context the governing equations are given in the form

$$\frac{d}{dt}\int_{\sigma(t)} \underline{u}\, d\underline{x} = -\sum_{j=1}^{3}\int_{\partial\sigma(t)} \underline{f}_j^c(\underline{u}) n_j\, ds + \frac{1}{\mathrm{Re}_\infty}\sum_{j=1}^{3}\int_{\partial\sigma(t)} \underline{f}_j^v(\underline{u}) n_j\, ds. \qquad (1)$$

Integration is performed on time-dependent control volumes $\sigma(t) \subset \mathbf{R}^3$ with outer unit normal vector $\underline{n}$. Here, $\underline{u} = (\rho, \rho v_1, \rho v_2, \rho v_3, \rho E)^T$ denotes the vector of conserved variables, $\underline{f}_j^c$ and $\underline{f}_j^v$ are the convective and viscous fluxes, respectively, given by

$$\underline{f}_j^c(\underline{u}) := \begin{pmatrix} \rho V_j \\ \rho v_1 V_j + \delta_1^j p \\ \rho v_2 V_j + \delta_2^j p \\ \rho v_3 V_j + \delta_3^j p \\ \rho H V_j + p v_{\mathrm{grid},j} \end{pmatrix},$$

$$\underline{f}_j^v(\underline{u}) := \begin{pmatrix} 0 \\ \tau_{1j} \\ \tau_{2j} \\ \tau_{3j} \\ \sum_{i=1}^{3} v_i \tau_{ij} + \frac{\mu\gamma}{\mathrm{Pr}_\infty}\partial_{x_j} e \end{pmatrix}.$$

The quantity e denotes internal energy which is given by $e = E - \frac{1}{2}(v_1^2 + v_2^2 + v_3^2)$ and the enthalpy H is defined as $H := E + p/\rho$. Pressure is given by the equation of state $p = (\gamma - 1)\rho\left(E - \frac{1}{2}(v_1^2 + v_2^2 + v_3^2)\right)$, γ being the ratio of specific heats. The temperature

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

is given by $T = \gamma(\gamma-1)\mathrm{Ma}_\infty^2 e$ and the elements of the shear stress tensor are $\tau_{ij} = \mu(\partial_{x_j} v_i + \partial_{x_i} v_j) + \delta_i^j \lambda(\partial_{x_1} v_1 + \partial_{x_2} v_2)$, with the viscosity assumed to follow the Sutherland law $\mu = T^{1.5}(1+S)/(T+S)$, where $S = 110° K/T_\infty$. Moreover, the connection between the termal conductivity and the viscosity is defined by Stokes' hypothesis to be $\lambda = -\frac{2}{3}\mu$.
The velocity $\underline{v}_{\text{grid}}$ is the velocity of the moving reference frame and $V_j := v_j - v_{\text{grid},j}$ denotes the contravariant velocity.

4 The DLR-$\mathcal{T}$-code

In order to simplify notation we describe the details of our numerical method in two space dimensions. The extension to three space dimensions follows by straightforward considerations based on the 2-d case.

4.1 Finite volume approximation

We consider conforming triangulations $\mathcal{T}_h$ consisting of tetrahedra (triangles in two-d) in the sense of Ciarlet [5] and define a discrete control volume $\sigma_i(t)$ as the volume of the barycentric subdivision of $\mathcal{T}_h$ enclosing the node $\underline{x}_i = (x_{i,1}, x_{i,2})^T$ and bounded by the straight line segments $l_{ij}^k, k = 1, 2$, connecting the midpoint of the edge with the point $\underline{x}_s$. The geometry of the control volumes is shown in figure 1. Figure 2 shows the boundary of a control volume and serves to define our notations. The point $\underline{x}_s$ is defined by

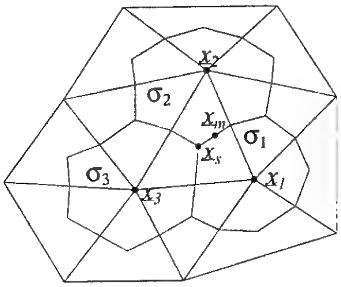


Figure 1: Control volumes in 2-d

$$\underline{x}_s := \sum_{m\in\{i,j,k\}} \alpha_m^s \underline{x}_m,$$

with

$$\alpha_m^s := \frac{1}{2(|l_i| + |l_j| + |l_k|)} \sum_{\substack{\overline{m}\in\{i,j,k\} \\ \overline{m}\neq m}} |l_{\overline{m}}|,$$

in order to account for highly stretched meshes in boundary layer regions.
Utilizing our notion of control volumes and denoting the cell average on $\sigma_i(t)$ by $\underline{u}_i(t) := \frac{1}{|\sigma_i|}\int_{\sigma_i(t)} \underline{u}(\underline{x}, t)\, d\underline{x}$, the Navier-Stokes equations (1) can be re-written in the form

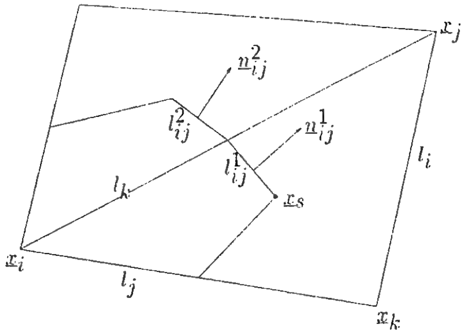


Figure 2: Boundary of a control volume in 2-d

$$\frac{d}{dt}\underline{u}_i(t) =$$
$$\frac{1}{|\sigma_i(t)|} \sum_{j\in N(i)} \sum_{k=1}^{2} \int_{l_{ij}^k} \sum_{\ell=1}^{2} \left(\frac{1}{\mathrm{Re}_\infty} \underline{f}_\ell^v(\underline{u}) - \underline{f}_\ell^c(\underline{u}) \right) n_\ell \, ds,$$

where $N(i) := \{j \mid \partial\sigma_j \cap \partial\sigma_i \neq \emptyset\}$ is the set of indexes of nodes neighbouring node $\underline{x}_i$. Since the line integrals are not defined if $\underline{u}$ is discontinuous two numerical flux functions are introduced, namely $\underline{H}, \underline{G} : \mathbf{R}^4 \times \mathbf{R}^4 \times \mathbf{R}^2 \to \mathbf{R}^4$, approximating the convective and viscous fluxes, respectively, and satisfying the fundamental consistency conditions

$$\underline{H}(\underline{u}, \underline{u}; \underline{n}) = \sum_{i=1}^{2} \underline{f}_i^c(\underline{u}) n_i, \quad \underline{G}(\underline{u}, \underline{u}; \underline{n}) = \sum_{i=1}^{2} \underline{f}_i^v(\underline{u}) n_i.$$

In our implementation the combined Riemann solver AUSMDV following Liou and Wada [33] is used for the numerical flux $\underline{H}$, which includes Hänel's scheme [10] and was extended in [18] for the use in an implicit formulation considering moving grids. Several other choices, like Roe's or Osher's Riemann solver, are easily implemented in the current framework. The viscous fluxes are discretised by the central difference

$$\underline{G}(\underline{u}_i, \underline{u}_j; \underline{n}) := \sum_{\ell=1}^{2} \underline{f}_\ell^v \left(\frac{\underline{u}_i + \underline{u}_j}{2} \right) n_\ell.$$

Applying the midpoint rule to the integral along l_{ij}^k results in

$$\frac{d}{dt}\underline{u}_i(t) =$$
$$\frac{1}{|\sigma_i|} \sum_{j\in N(i)} \sum_{k=1}^{2} \left\{ \frac{1}{\mathrm{Re}_\infty} \underline{G}\left(\hat{\underline{u}}_i(\underline{x}_{ij}^k, t), \hat{\underline{u}}_j(\underline{x}_{ij}^k, t); \underline{n}_{ij}^k(t)\right) \right.$$
$$\left. - \underline{H}\left(\hat{\underline{u}}_i(\underline{x}_{ij}^k, t), \hat{\underline{u}}_j(\underline{x}_{ij}^k, t); \underline{n}_{ij}^k(t)\right) \right\} |l_{ij}^k|$$
$$+\mathcal{O}(h^2) + \mathcal{O}(h^q),$$

where the first error term is due to the quadrature rule while the second error term depends on the functions $\hat{\underline{u}}$. Using cell average values, i.e. $\hat{\underline{u}}_i \equiv \underline{u}_i$, results in a first order approximation, i.e. $q = 1$, due to the weak approximation property of the cell average operator, see [22], [23]. To increase the approximation order a recovery function $\hat{\underline{u}}_i$ is sought on σ_i which approximates $\underline{u}$ at least with order $\mathcal{O}(h^2)$. It is easily seen that linear polynomials recover $\underline{u}$ up to this order.

4.2 Recovery algorithms

If $\underline{x}_b^i$ denotes the barycentre of the control volume σ_i then a linear polynomial

$$\hat{\underline{u}}_i(\underline{x},t) = \sum_{|\underline{\alpha}|\leq 1} a_{\underline{\alpha}}(t)(\underline{x}-\underline{x}_b^i) \tag{2}$$

has to be recovered on $\sigma_i(t)$ such that it satisfies the recovery condition

$$\frac{1}{|\sigma_i(t)|}\int_{\sigma_i(t)} \hat{\underline{u}}_i \, d\underline{x} = \underline{u}_i .$$

Recovery in box-type methods is best described in terms of a meta-triangulation $\tilde{T}_i^h$ which is defined to be the triangulation of the barycentres of control volume σ_i and the surrounding boxes σ_{i_k} if $\underline{x}_b^i$ is connected with each of the surrounding $\underline{x}_b^{i_k}$, see figure 3. If $\tilde{E}(i) := \{\tilde{T} \in \tilde{T}_i^h\}$ denotes the set of the meta-triangles surrounding $\underline{x}_b^i$ then a linear polynomial $\underline{\pi}_{\tilde{T}}$ can be computed on each of the $\tilde{T}$.

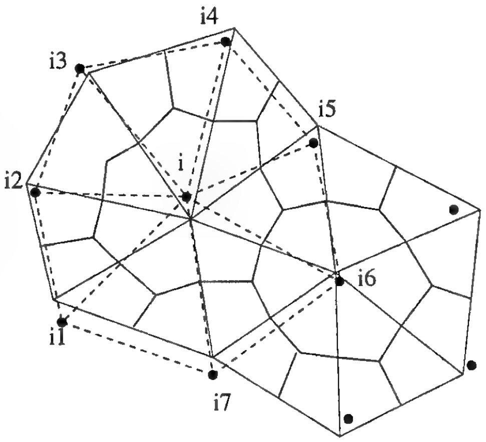


Figure 3: Meta-triangulation of the barycentres

In a TVD-like approach, compare [24], the gradient $(a_{10}^i, a_{01}^i)^T$ of the recovery polynomial (2) can be obtained from the linear interpolants in a completely isotropic manner, namely

$$(\underline{a}_{10}^i, \underline{a}_{01}^i)^T := \frac{1}{|\sigma_i|}\sum_{\tilde{T}}\int_{\sigma_i\cap\tilde{T}} \nabla\underline{\pi}_{\tilde{T}} \, d\underline{x}.$$

Then, defining $\underline{a}_{00}^i := \underline{u}_i$, the polynomial (2) will certainly satisfy the recovery condition. However, the isotropic recovery of the gradients does not take care of shocks in the solution and will thus lead to instabilities. According to the TVD methodology a slope limiter Φ_i hast to be introduced such that the recovery polynomial is written in the form

$$\hat{\underline{u}}_i(\underline{x},t) = \underline{u}_i + \Phi_i(a_{10}^i, a_{01}^i)^T \cdot (\underline{x}-\underline{x}_b^i).$$

We have good experience in using the limiter described by Barth and Jesperson in [4], but convergence to steady state is enhanced if one adds a modification as suggested by Venkatakrishnan in [31].
A simple ENO-type recovery can also be described in terms of the linear interpolants $\underline{\pi}_{\tilde{T}}$. The linear recovery polynomial $\hat{\underline{u}}_i$ on the box σ_i is then chosen to be the one linear polynomial $\underline{\pi}_{\tilde{T}^*}$ on the surrounding meta-triangles for which the modulus of the gradient is minimal, i.e. for which

$$\left|\nabla\pi_{\tilde{T}^*,i}\right| = \min_{\tilde{T}}\left|\nabla\pi_{\tilde{T},i}\right|$$

is valid where $\nabla\pi_{\tilde{T}^*,i}$ denotes the i-th component. Experience with this type of recovery is reported in [22] and [25].
In order to further increase the spatial accuracy of the DLR-τ-code we are currently working on the extension towards a third order scheme by recovering quadratic polynomials close to the ideas of Abgrall, see [1], [2]. In [3] an algorithm based on Mühlbach expansions was developed which allows the efficient and stable computation of quadratic recovery polynomials in a step-by-step manner. Preliminary numerical results concerning a third-order τ-code are given in [15].

4.3 Optimal recovery

Although polynomial recovery functions seem to be attractive at first glance for their simplicity, their main drawback lies in the enormous widening of the stencil if higher order recoveries are sought. On the other hand, even locally defined polynomials of high degree exhibit weak properties concerning their oscillatory behaviour. Additionally, as can be seen from application of the theory of Optimal Recovery as reviewed in [22], polynomials do not exhibit any optimality condition with respect to their recovery properties. We do want to recover a function $\hat{\underline{u}}_i$ on σ_i for which the difference

$$\left|\hat{\underline{u}}_i(\underline{y},t) - \underline{u}(\underline{y},t)\right|$$

between the recovery function and the true solution at the Gauss points $\underline{y}$ is smallest. Functions minimizing semi-norms in their associated function spaces (i.e. Splines) are exactly those functions for which the above quantity is minimal. In multiple space dimensions splines are found in the class of radial basis functions, for example the well-known thin-plate spline. First experiments with this kind of recovery functions in [22], [23] showed impressive increase in accuracy. Although recovery of radial basis functions is much too expensive as compared to polynomial recovery the techniques developed in [3] could very well provide a framework in which these more complicated functions could be competitive with polynomial algorithms.

In recovery with radial basis functions a recovery function of the form

$$\hat{u}_{i,l}(\underline{x},t) = \sum_{j=0}^{N-1} \lambda_j \mathsf{A}(\sigma_i)^{\underline{y}} \Phi(|\underline{x}-\underline{y}|,t) + \sum_{k=1}^{M} \beta_k \pi_k(\underline{x},t)$$

is sought for the l-th component, where $\mathsf{A}(\sigma_i)^{\underline{y}} f := \frac{1}{|\sigma_i|}\int_{\sigma_i} f(\underline{y})\, d\underline{y}$ denotes the cell average operator. The radial function Φ is assumed to satisfy the fundamental condition of being conditionally positive definite and $\pi_k, k = 1, \ldots, N$, denote a basis of the space of polynomials of a certain degree, which depends on the radial function Φ chosen. The number of nodes N in the recovery stencil is another quantity which has to be choosen. Numerical experience gained so far has indicated that polynomial-based ENO stencil selection criteria work well also in the case of radial basis functions.

Using the well known thin plate spline

$$\begin{aligned}\Phi(\underline{x},t) \quad = \quad & \sum_{j=0}^{N-1} \lambda_j \mathsf{A}^{\underline{y}}(\sigma_j) \left(|\underline{x}-\underline{y}|^2 \log(|\underline{x}-\underline{y}|)\right. \\ & \qquad\qquad + \sum_{\underline{\alpha} \leq 2} a_{\underline{\alpha}} \underline{x}^{\underline{\alpha}},\end{aligned}$$

which, by construction, is able to reproduce linear polynomials, amounts to use at least four control volumes in the stencil. In an ENO-like manner one can think of the stencil selection according to figure 4, where the control volumes were chosen to be triangles.

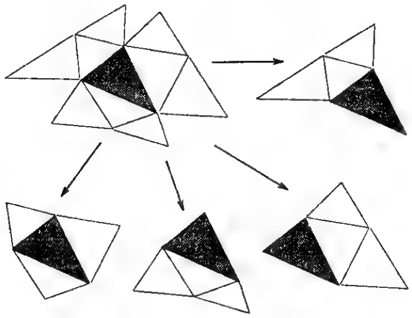


Figure 4: The construction of four node sets out of a certain neighbourhood

If on each of the four stencil sets a radial basis recovery function is computed the one with the smallest total variation norm is selected and assigned to the control volume. First results of this procedure are reported in section 6.

Meanwhile, radial basis functions with compact support are being constructed. We mention the class of Wu functions as designed in [36] and the very recent developments of Wendland [34]. These functions are unconditionally positive definite and thus do not need the polynomial augmentation as the thin plate spline. Furthermore, their compact support makes them very attractive for practical purposes. Whether these functions can be competitive in runtime to polynomial-based recovery algorithms is the contents of future research on ENO approximations.

4.4 Time stepping schemes and parallelism

The DLR-τ-code was originally supplied with an explicit Runge-Kutta time stepping algorithm designed by Shu and Osher in [21] which respects the TVD-properties of the spatial discretisation, see [24]. However, these schemes are limited in CFL number by 1 which is a dramatic upper bound for application in an adaptive framework where grid cells can be very small. In the meantime other Runge-Kutta schemes with up to five stages are in use and show satisfying behaviour especially when used in a multigrid environment for steady problems. For the computation of unsteady flows, as pitching airfoils, the restrictions due to the CFL condition are still too strong. One way to overcome the limitations of explicit time stepping schemes is the use of parallel computers which is easy in the case of finite volume approximations because domain decomposition is natural. A grid partitioner was developed in connection with an intelligent load balancing algorithm to re-divide and re-distribute grid patches depending on the load of the processors used. In that framework a parallel computation is easily done in an environment consisting of a cluster of workstations running PVM or P4 while the machines are still occupied by other users.

In figure 5 the grid of a channel with forward facing step is shown. The grid partitioner has divided the grid into 59 patches which contain nearly the same number of nodes. The possible speedup is documented in the diagram of figure 6 where speedup vs. num-

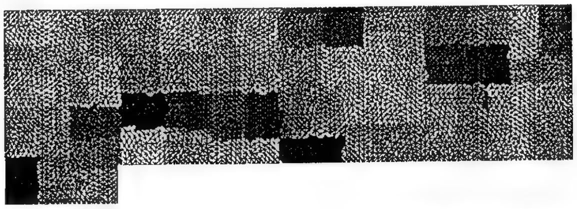


Figure 5: Grid partitioning

ber of processors is shown for an Intel PARAGON. The flow is the supersonic test case by Woodward and Colella [35] as discussed also in section 6.1. As can be seen from figure 6 the present approach towards parallelism through domain decompostion leads to a very efficient method.

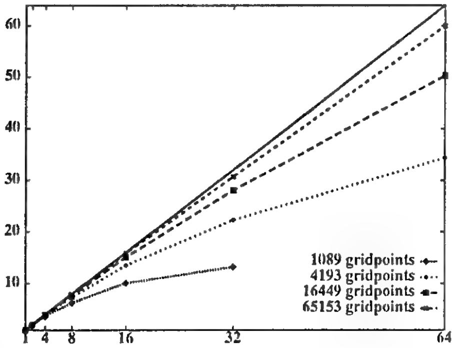


Figure 6: Speedup on an Intel PARAGON

For use on conventional machines an implicit time stepping scheme according to

$$\begin{aligned}\frac{d\underline{u}}{dt}\left(t^{n+1}\right) &= g_1(\eta,\phi)\frac{\underline{u}\left(t^{n+1}\right)-\underline{u}\left(t^{n}\right)}{t^{n+1}-t^{n}} \\ &\quad - g_2(\eta,\phi)\frac{\underline{u}\left(t^{n}\right)-\underline{u}\left(t^{n-1}\right)}{t^{n}-t^{n-1}} \\ &\quad + O\left(\left(t^{n+1}-t^{n}\right)^{\phi+1}\right),\end{aligned}$$

where $\eta = \frac{t^{n+1}-t^{n}}{t^{n}-t^{n-1}}$ and

$$\begin{aligned} g_1(\eta,\phi) &= \begin{cases} \frac{2+\eta}{1+\eta} & \text{if} \quad \phi = 1 \\ 1 & \text{if} \quad \phi = 0 \end{cases} \\ g_2(\eta,\phi) &= \begin{cases} \frac{1}{\eta+\eta^2} & \text{if} \quad \phi = 1 \\ 0 & \text{if} \quad \phi = 0 \end{cases} ,\end{aligned}$$

was designed.

The numerical flux functions are evaluated at the time '$n+1$' whereby a linearisation is necessary which leads to a linear system of equations in the form

$$\underline{\underline{A}}_{ii}\Delta\underline{u}_i^n + \sum_{j\in N(i)} \underline{\underline{B}}_{ij}\Delta\underline{u}_j^n = \underline{b}_i \quad , \quad i = 1,\ldots,I \quad ,$$

where $\Delta\underline{u}_i^n = \underline{u}_i^{n+1} - \underline{u}_i^n$ and $\underline{\underline{A}}_{ii}, \underline{\underline{B}}_{ij} \in \mathbf{R}^{4\times4}$. Consequently, for each time step a linear system

$$\underline{\underline{A}}\,\underline{y} = \underline{b} \qquad (3)$$

has to be solved, where $\underline{\underline{A}}$ is a large sparse nonsymmetric matrix. For the solution of the system (3) the GMRES algorithm developed by Saad and Schulz [19], [20] is used. Therefore, the system is transformed into an equivalent minimisation problem. First, we define the function $f : \mathbf{R}^n \to \mathbf{R}_0^+$ by $f(\underline{y}) = \|\underline{b} - \underline{\underline{A}}\,\underline{y}\|_2^2$ and choose an arbitrary initial vector y_0. Starting with $m = 0$ the residual $\overline{r}_m = \min_{\substack{\underline{y}=\underline{y}_0+\underline{z} \\ \underline{z}\in K_m}} f(\underline{y})$ is computed, where $K_m(\underline{\underline{A}}, \underline{r_0}) := span\left\{\underline{r_0}, \underline{\underline{A}}\,\underline{r_0}, \underline{\underline{A}}^2\underline{r_0}, \ldots, \underline{\underline{A}}^{m-1}\underline{r_0}\right\}$ denotes the m-th Krylov subspace and $\underline{r_0} = \underline{b} - \underline{\underline{A}}\,\underline{y_0}$. Now we increase m until $\overline{r}_m$ is below a given tolerance. Then we compute the optimal approximate solution $\underline{y}_m = \arg \min_{\substack{\underline{y}=\underline{y_0}+\underline{z} \\ \underline{z}\in K_m}} f(\underline{y})$. Considering the fact that the expense to calculate the residual increases with the Krylov subspace dimension it is efficient to limit this dimension. If this limit is reached before that of the tolerance the approximation $\underline{y}_m$ has to be calculated and used as the initial value during a repetition. This technique is called "GMRES with restart".

Since the convergence rate of an iterative method depends on the condition number of the matrix $\underline{\underline{A}}$, an incomplete LU-factorisation is used as a preconditioner in order to decrease the condition number. Hereby the incomplete LU-factorisation is a pair of a lower left $(\underline{\underline{L}})$ and a upper right $(\underline{\underline{U}})$ matrix satisfying the following three conditions:

1. $\underline{\underline{U}}_{ii}$ presents the unit matrix for all i,
2. $\underline{\underline{L}}_{ij} = \underline{\underline{U}}_{ij} = \underline{\underline{A}}_{ij}$, if $\underline{\underline{A}}_{ij}$ is a null matrix,
3. $\left(\underline{\underline{LU}}\right)_{ij} = \underline{\underline{A}}_{ij}$, if $\underline{\underline{A}}_{ij}$ is not a null matrix

and the linear system is transformed into

$$\underline{\underline{A}}\,\underline{\underline{U}}^{-1}\underline{\underline{L}}^{-1}\underline{\tilde{y}} = \underline{b} \quad , \quad \underline{y} = \underline{\underline{U}}^{-1}\underline{\underline{L}}^{-1}\underline{\tilde{y}}.$$

A detailed description of these preconditioned GMRES algorithm in comparison with other implicit and explicit finite volume schemes is presented in [17].

5 Adaptive concepts

5.1 Refinement algorithms

Over the years experience was gained with several different refinement/coarsening strategies for triangulations. This work is documented in [26], [27], [11] and [13]. Numerical experience indicated that, at least for reliable Euler grids, a version of the isotropic red-green refinement as described in [14] gives superior grids. In this refinement strategy triangles which have to be refined are red-refined according to figure 7. Remaining triangles with two hanging nodes are also red-refined before green refinement turns the triangulation again into a conforming one. Note that at the beginning of each refinement cycle the previous green refinements are removed in order to keep the triangulations stable, i.e. in order to avoid too small angles occuring after several adaptation cycles.

In a corresponding re-coarsening strategy several topological configurations can be identified in which points can be removed from the grid. It can be

shown, see [14], that a refined mesh can always be completely coarsened up to its initial state.
Note that in order to keep the process of coarsening conservative it is necessary to use interpolation procedures respecting conservation. Examples of such procedures are given in [14].

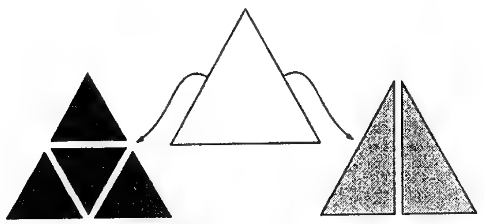


Figure 7: Red (left) and green refinement of a triangle

5.2 Error indicators

In contrast to classical approaches in CFD the DLR-τ-Code relies on residual-based error indicators which were developed in subsequent papers [26], [27], [11], [28]. This type of indicators was developed for use in codes for the Euler equations but we are currently working on extensions towards the Navier-Stokes equations. If $\mathcal{L}\underline{u} = \underline{0}$ denotes the abstract form of the Euler equations in which $\mathcal{L}$ is the corresponding differential operator of first order and $\underline{\pi}_T$ denote the linear interpolants of the flow variables on triangle T, then the local error of the numerical method under consideration is defined by $\underline{e}_T^h := \underline{\pi}_T - \underline{u}$. If the numerical approximation $\underline{\pi}_T$ is inserted into the differential equation the deviation from the zero vector is a measure of closeness to the exact solution. The quantity

$$\underline{r}_T^h := \mathcal{L}\underline{\pi}_T$$

is therefore called the residual. It was shown in [28] that a two-sided error bound of the form

$$C_1 ||\underline{r}_T^h||_{D^*(T)} \leq ||\underline{e}_T^h||_{L^2(T)} \leq C_2 ||\underline{r}_T^h||_{D^*(T)}$$

can be proved where $||\cdot||_{D^*(T)}$ denotes the dual graph-norm. First numerical results were presented in [28] which indicated that the use of the dual graph-norm leads to similar results as the use of the weighted L^2-norm

$$h||\underline{r}_T^h||_{L^2(T)},$$

h denoting the length of the longest edge of T, which was used for heuristic reasons before, see [11], [26], [27]. In [30] Süli was able to prove that the dual graph-norm is indeed essentially equivalent to $h||\underline{r}_T^h||_{L^2(T)}$ and since this locally weighted L^2-norm is much easier to implement this is the error indicator of our choice.
The additional problem occuring with the Navier-Stokes equations lies in the second derivatives inherent in the diffusive fluxes. Although we are currently not able to prove error bounds it seems possible to approximate the second derivatives in the computation of the residual in a measure-theoretic way by sampling the jumps of the first derivatives across the edges in normal direction. This type of error indicators was inspired by the work of C. Johnson et al. on the adaptive streamline diffusion finite element method, see [7], [12], and developed by Göhner and Warnecke [9]. We are currently investigating this type of indicators for compressible flow [29].

6 Numerical results

6.1 Unsteady flow in a channel

To show the ability of the code to adaptively resolve the flow features we consider the test case of $\mathrm{Ma}_\infty = 3$ flow through a channel with forward facing step. This case was used by Woodward and Colella [35] for extensive comparison of finite difference schemes. In figure 8 four grids at consecutive times are shown together with the corresponding density distributions. As can be seen the adaptive algorithm has not only resolved all of the flow features but also succeeded in coarsening those part of the meshes which were previously refined. The computation was done with the parallel τ-Code on a cluster of workstations using PVM.

6.2 Pitching airfoil

Figures 9 and 10 show the results of a calculation of an unsteady inviscid flow about the NACA0012 airfoil in comparison with experimental data. In this case the airfoil is pitching harmonically about the quarter chord point with a reduced frequency of $k = 0.1628$ and an amplitude of $\alpha = 2.51°$. The freestream Mach number is $\mathrm{Ma}_\infty = 0.755$ and the angle of attack initially is $0.016°$. Consequently, the time-dependent angle of attack is $\alpha(t) = 0.016° + 2.51° \sin(0.1628 \cdot t)$.
Figure 9 shows the obtained instantaneous pressure distribution in comparison with the experimental data for several times during the third cycle of motion. Figure 10 shows the comparison of the lift coefficient and the moment coefficent vs. the time-dependent angle of attack. The computational data are very close to the experimental ones. Note, that no diffusive effects were included in this calculation.

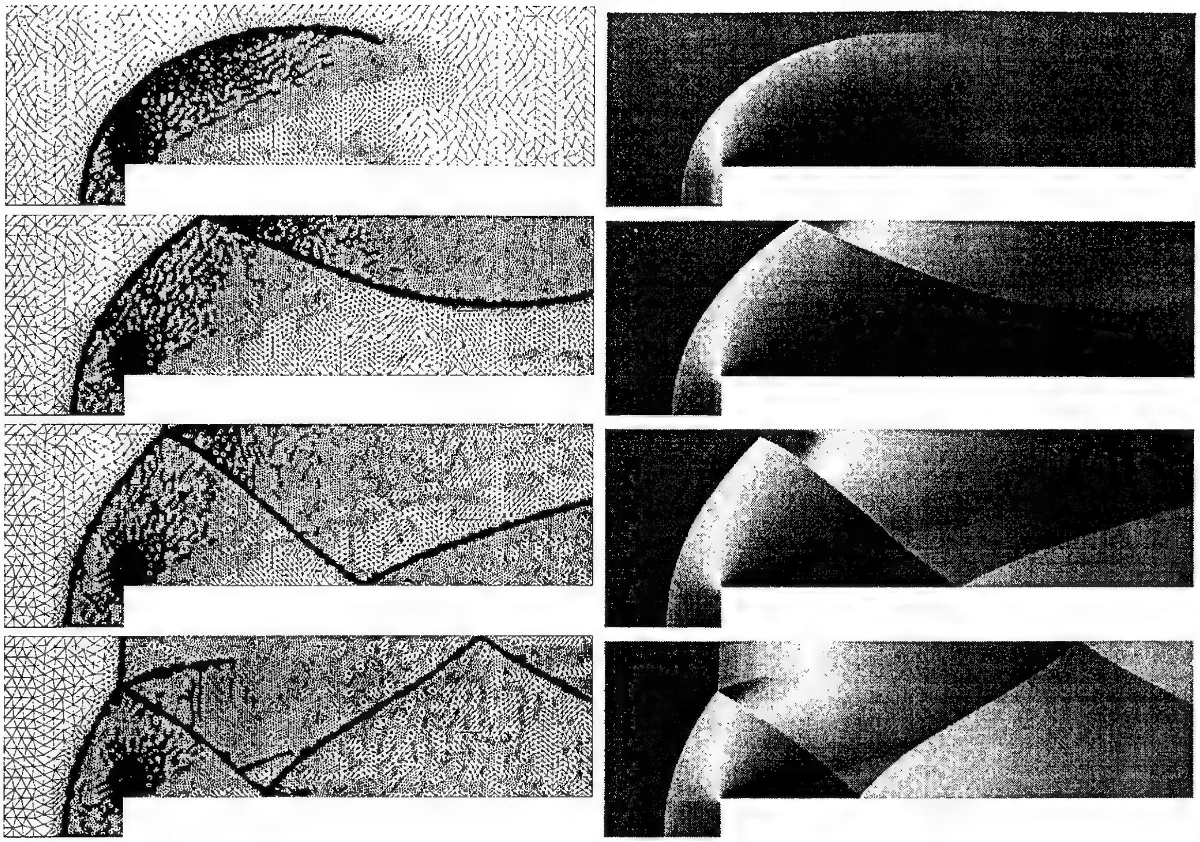


Figure 8: Evolution of the adaptive grid and corresponding solutions

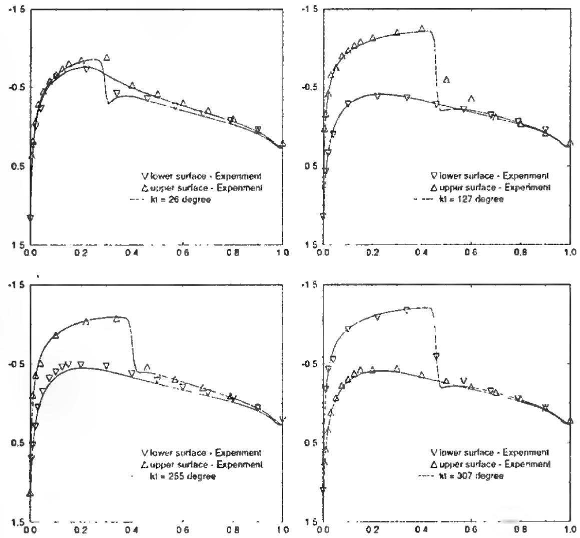


Figure 9: Comparison of the instantaneous pressure distribution between the numerical computation (inviscid) and experimental data

6.3 Viscous flow about a NACA0012 airfoil

The next case was chosen to test the method and the adaptation algorithm for viscous flow computations. We consider the steady laminar flow about a NACA0012 airfoil with a Reynolds number of 500, a Prandtl number of 0.72, a reference temperature of 273 degree Kelvin, a freestream Mach number of 0.85 and an angle of attack of 0°. The obtained Mach number distribution is shown in figure 11 and figure 12 presents the adapted grid. The adaptation indicator used is based on the finite element residual of the Navier-Stokes equations.

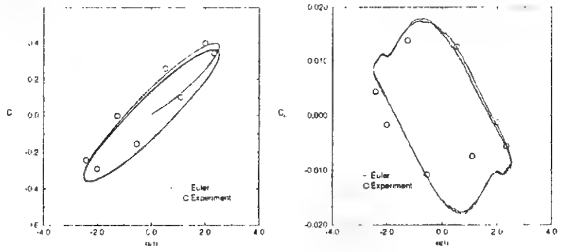


Figure 10: Comparison of lift and momentum coefficient vs. time-dependent angle of attack between the numerical computation (inviscid) and experimental data

6.4 3-D transonic wing

As a three dimensional testcase the inviscid flow about an Onera M6 wing with $\mathrm{Ma}_\infty = 0.84$ and

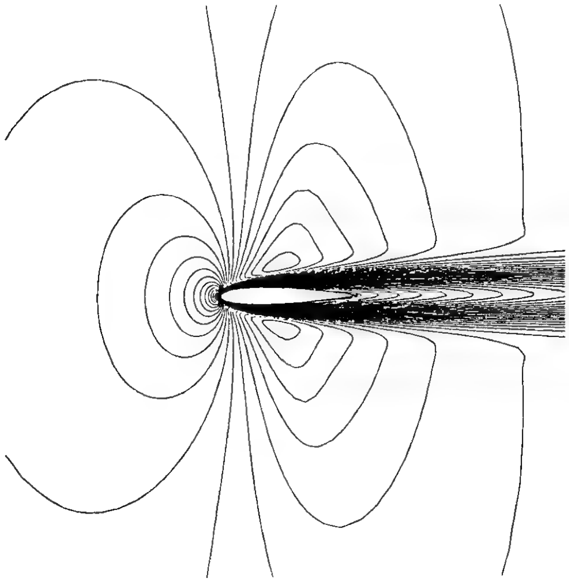


Figure 11: NACA0012 airfoil - Mach number distribution.

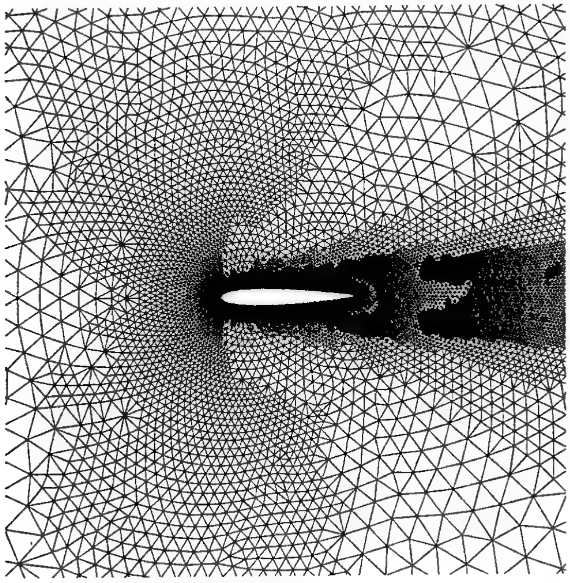


Figure 12: NACA0012 airfoil - Adapted grid.

an angle of attack of 3.06° is considered. Figure 13 shows isolines of the Mach number distribution on a coarse hybrid grid with less than 40,000 gridpoints. Figure 14 shows the distribution of the computed L_2-norm of the finite element residual for the same solution. It can be seen that the much too coarse resolved leading and trailing edge, the tip region as well as the shock were picked up. Thus, also in three space dimensions the finite element residual of the Euler equations can be used as an adaptation indicator.

To accelerate convergence to steady state agglomeration multigrid as described by Venkatakrishnan and Mavriplis [32] is used. The coarse grid discretisation is done like described in [8]. In figures 15 and 16 the advantage of the multigrid solver over the single grid solver is shown. As one can see in figure 16, the lift coefficient in the single grid computation is oscillating with about 5 to 10 percent after 2500 timesteps while the lift coefficient of the multigrid computation is converged already after a CPU time that corresponds to one hundred single grid timesteps.

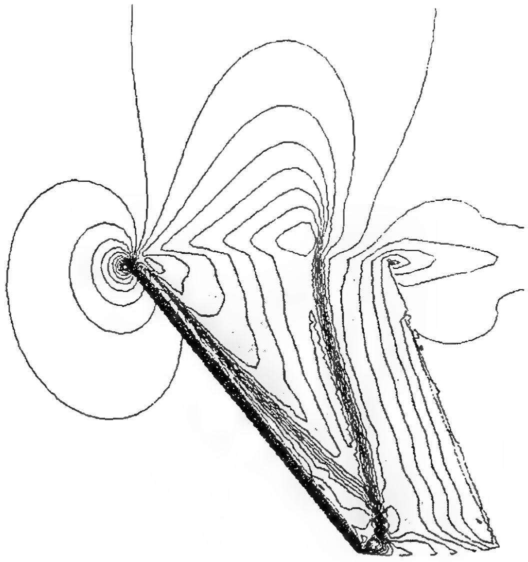


Figure 13: Isolines of the Mach number distribution.

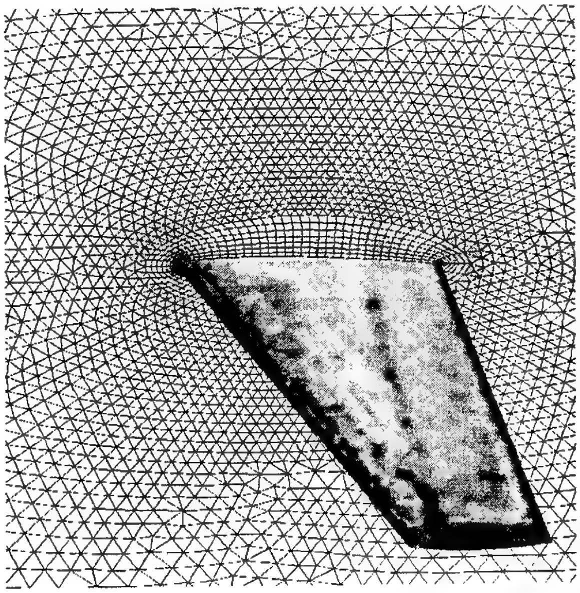


Figure 14: L_2-Norm of the residual displayed on the surface. Darker regions indicate heigher L_2-norm.

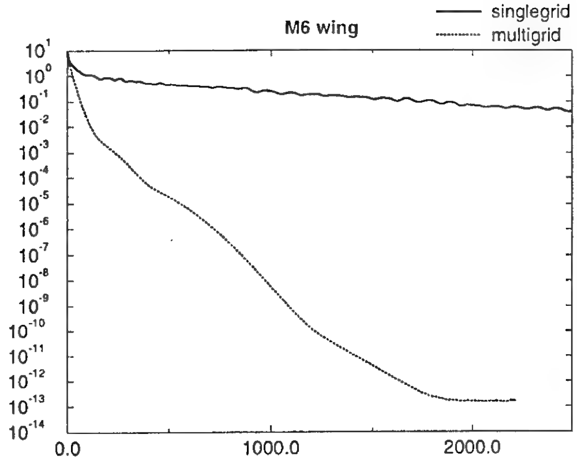


Figure 15: L_2-Norm of the density residual vs. CPU-time.

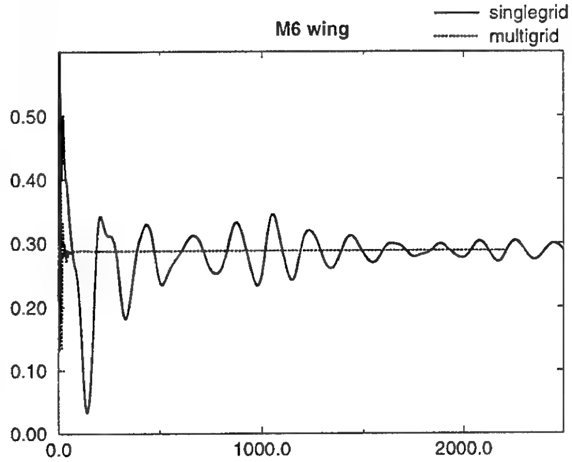


Figure 16: Lift coefficient vs. CPU-time.

6.5 Radial recovery functions

The accuracy of recovery algorithms based on radial basis functions can be seen in an application to a simple model problem. Consider the linear partial differential equation

$$\begin{aligned} \partial_t u + \underline{a} \cdot \nabla_{\underline{x}} u &= 0 \\ u(\underline{x}, 0) &= u_0(\underline{x}), \quad \underline{x} \in \mathbf{R}^2 \end{aligned}$$

with

$$\underline{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} := \begin{pmatrix} \frac{1}{2} - x_2 \\ x_1 - \frac{1}{2} \end{pmatrix}$$

and

$$u_0(\underline{x}) = \begin{cases} -\frac{1}{0.01} R + 1 & ; \quad R \leq 0.01 \\ 0 & ; \quad \text{else.} \end{cases}$$

where $R := (x_1 - \frac{3}{4})^2 + (x_2 - \frac{1}{2})^2$. The initial function is a cone of unit height which is rotated around the origin under the action of the differential equation. Measuring the remaining cone height after 180° of rotation gives a reliable criterion concerning the accuracy of the recovery. In figure 17 the grid used in shown with a numerical solution of a finite volume approximation without recovery. The remaining cone height is a disappointing 0.382 and the shape of the cone is dramatically corrupted. Using a linear polynomial recovery algorithm following Durlofsky, Engquist, Osher [6] results in the solution shown in the left part of figure 18. The cone height now is 0.635 but the shape of the cone is still lacking regularity. Using the thin plate spline recovery as described in 4.3 results in a cone with proper shape and height 0.886. This solution is shown in the right part of figure 18.

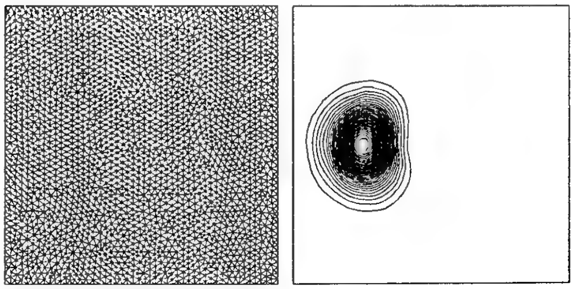


Figure 17: Grid and solution without recovery

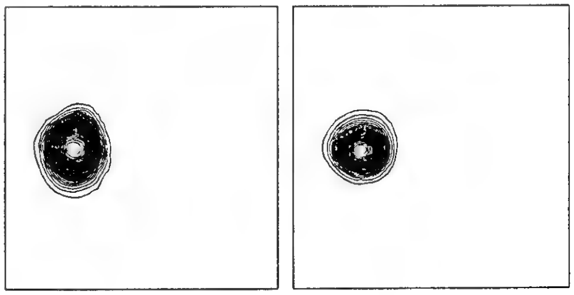


Figure 18: Solutions with linear polynomial (left) and thin plate spline recovery

Experiments with radial basis functions with compact support have indicated even better numerical results than those obtained with the thin plate spline. Additionally, methods for the fast construction of recovery functions are currently being developed [3] so that there is hope that these functions can be implemented in the DLR-τ-Code in the near future.

References

[1] R. Abgrall - On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation. *J. Comp. Phys.* **114**, *45-58, (1994)*

[2] R. Abgrall - Design of an essentially nonoscillatory reconstruction procedure on finite-element-

type meshes. *ICASE Report No.91-84, (1991).* Revised version: *INRIA Report No 2342, (1994)*

[3] R. Abgrall, Th. Sonar - On the use of Mühlbach expansions in the recovery step of ENO methods. *DLR Interner Bericht IB 223-95 A 34, (1995)*

[4] T.J. Barth, D.C. Jespersen - The design and application of upwind schemes on unstructured meshes. *AIAA paper 89-0366, (1989).*

[5] P.G. Ciarlet - The finite element method for elliptic problems. *North-Holland, 2nd edt. (1987)*

[6] L.J. Durlofsky, B. Engquist, S. Osher - Triangle Based adaptive stencils for the solution of hyperbolic conservation laws. *J. Comp. Phys.* **98**, *64-73, (1992)*

[7] K. Eriksson, C. Johnson. Adaptive finite element methods for parabolic problems I. A linear model problem. *Chalmers Univerity of Technology, Department of Mathematics, preprint 88:31 (1988).*

[8] M. Galle - Solution of the Euler and Navier-Stokes equations on hybrid grids. *This volume, paper no. 30*

[9] U. Göhner, G. Warnecke - A second-order finite difference error indicator for adaptive transonic flow computations. *Num. Math.* **70**, *129-161, (1995)*

[10] D. Hänel, R. Schwane - An implicit flux-vector splitting scheme for the computation of viscous hypersonic flow. *AIAA paper 89-0274 (1989)*

[11] V. Hannemann, D. Hempel, Th. Sonar - Adaptive computation of compressible flow fields with the DLR-τ-code. *in: Numerical Methods for the Navier-Stokes Equations, F.-K. Hebeker, R. Rannacher, G. Wittum (Eds.), Notes on Numerical Fluid Mechanics, Volume 47, Vieweg Verlag, 101-110, (1994)*

[12] P. Hansbo, C. Johnson - Adaptive streamline diffusion methods for compressible flow using conservation variables. *Comp. Methods Appl. Mech. and Engrg.* **87**, *267-280, (1991).*

[13] D. Hempel - Dynamic adaption of triangular grids. *DLR Interner Bericht IB 223-95 A 38, (1995)*

[14] D. Hempel - Isotropic refinement and recoarsening in 2 dimensions. *DLR Interner Bericht IB 223-95 A 35, (1995)*

[15] D. Hietel, A. Meister, Th. Sonar - On the comparison of two different implementations of an implicit third-order ENO scheme of box type for the computation of unsteady compressible flow. *in preparation*

[16] D.J. Mavriplis - A three dimensional multigrid Reynolds-averaged Navier-Stokes solver for unstructured meshes. *ICASE report no. 94-29 (1994)*

[17] A. Meister - Ein Beitrag zum DLR-τ-Code: Ein explizites und implizites Finite-Volumen-Verfahren zur Berechnung instationärer Strömungen auf unstrukturierten Gittern. *DLR Interner Bericht IB 223-94 A 36, Göttingen, (1994)*

[18] A. Meister - Development of an implicit finite volume scheme for the computation of unsteady flow fields on unstructured moving grids. *to appear in Proceedings of the ICFD Conference on Numerical Methods for Fluid Dynamics, Oxford, (1995)*

[19] Y. Saad, M. H. Schulz - GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.* **7**, *856-869, (1986)*

[20] Y. Saad - Krylov subspace techniques, conjugate gradients, preconditioning and sparse matrix solvers. *von Karman Institute of Fluid Dynamics, Lecture series 1994-05, (1994)*

[21] C.-W. Shu, S. Osher - Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comp. Phys.* **77**, *439-471, (1988)*

[22] Th. Sonar - Multivariate Rekonstruktionsverfahren zur numerischen Lösung hyperbolischer Erhaltungsgleichungen. *Habilitationsschrift, TH Darmstadt, (1995). Also: DLR Forschungsbericht 95-02, (1995)*

[23] Th. Sonar - Optimal recovery using thin plate splines in finite volume methods for the numerical solution of hyperbolic conservation laws. *DLR Interner Bericht IB 223-94 A 42, (1994)*

[24] Th. Sonar - On the design of an upwind scheme for compressible flow on general triangulations. *Numerical Algorithms* **4**, *135-149, (1993)*

[25] Th. Sonar - On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: Polynomial recovery, accuracy, and stencil selection. *submitted: Journal of Computational Physics, (1995)*

[26] Th. Sonar - Strong and weak norm refinement indicators based on the finite element residual for compressible flow computation. *Impact of Computing in Science and Engincering* 5, *111-127, (1993)*

[27] Th. Sonar, V. Hannemann, D. Hempel - Dynamic adaptivity and residual control in unstaedy compressible flow computation. *Mathematical and Computer Modelling* 20, *201-213, (1994)*

[28] Th. Sonar, E. Süli - A dual graph norm refinement indicator for the DLR-τ-code. *DLR Forschungsbericht 94-24, (1994)*

[29] Th. Sonar, G. Warnecke - On a finite difference error indicator for adaptive approximations of conservation laws. *in preparation*

[30] E. Süli - A posteriori error analysis and global error control for adaptive finite element approximations of hyperbolic problems. *16th Int. Conf. on Num. Anal., Dundee, June (1995)*

[31] V. Venkatakrishnan - On the accuracy of limiters and convergence to steady state solutions. *AIAA paper 93-0880 (1993)*

[32] V. Venkatakrishnan, D.J. Mavriplis - Agglomeration multigrid for the three-dimensional euler equations. *ICASE report no. 94-5 (1994)*

[33] Y. Wada, M.-S. Liou - A flux splitting scheme with high-resolution an robustness for discontinuities. *ICOMP-93-50, (1993)*

[34] H. Wendland - Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Manusktript, Inst. Num. Ang. Math., Universität Göttingen, (1994)*

[35] P. Woodward, P. Colella - The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comp. Phys.* 54, *115-173, (1984)*

[36] Z.-M. Wu - Multivariate compactly supported positive definite radial basis functions. *Manuskript, Inst. Num. Ang. Math., Universität Göttingen, (1994)*

PARAMETRIC STUDIES OF A TIME-ACCURATE FINITE-VOLUME EULER CODE IN THE NWT PARALLEL COMPUTER

L. P. Ruiz-Calavera
INTA, Aerodynamics Division, Fluid Dynamics Department,
Carretera de Ajalvir Km. 4.5, 28850 Torrejon de Ardoz
SPAIN

N. Hirose
NAL, Computational Science Division,
7-44-1 Jindaiji-Higashi, Chofu-shi, Tokyo 182,
JAPAN

SUMMARY

A code to calculate unsteady aerodynamic loads on non-uniformly moving 3-D isolated wings has been prepared. The Euler equations are solved by means of a time-accurate Finite-Volume method with second order central spatial discretization and Runge-Kutta time integration. The code has been implemented in a parallel supercomputer. The numerical scheme used together with some representative results are presented.

LIST OF SYMBOLS

c = local chord
c_0 = reference length
C_l = section lift coefficient= $lift/q_\infty c\alpha_1$
c_p = pressure coefficient = $(p-p_\infty)/q_\infty\alpha_1$
d = dissipative flux
D = dissipative operator
E = specific total energy
F,G,H = components of Euler flux vector
k = reduced frequency = $\omega c/2V_\infty$
$k^{(2)},k^{(4)}$ = artificial viscosity constants
L = scaling factor
$\underline{M}_\infty$ = free-stream Mach number
n = surface normal unit vector
p = static pressure
Q = convective operator
q_∞ = free-stream dynamic pressure
R = residual
R^* = averaged residual
S = cell face area
t = time
t^* = dimensionless time = tV_∞/c_0
U = vector of conservative variables
u,v,w = components of flow velocity
V_∞ = free stream velocity
V_Σ = mesh velocity
α_0 = mean angle of attack
α_1 = pitching motion amplitude
β = Runge-Kutta coefficients
Σ = cell boundary
ϵ = residual averaging parameter
$\epsilon^{(2)},\epsilon^{(4)}$= artificial viscosity parameters
λ,μ,σ = spectral radius of flux Jacobian matrices in ξ, η, and ζ directions
ρ = air density
ξ,η,ζ = curvilinear coordinates
υ = shock wave sensor
Ω = cell volume
ω = frequency of oscillation

Subscripts
i = cell column
j = cell row
k = cell plane
n = cell face
q = Runge-Kutta stage

1. INTRODUCTION

Aeroelastic problems appear to be of increasing importance in the design of aircraft. The size of the structures and its elastic behavior, the aerodynamic interference of different components, transonic effects, structural and control nonlinearities, etc, are becoming a severe limiting factor. There is thus a strong need to apply sophisticated and reliable aeroelastic simulation tools already in the early design stage of a new development. These tools have to couple highly accurate, robust and user friendly CFD codes with Structural Dynamics software. Whereas the latter is already well established, the former still need development before a generally recognized standard code is available.

To clear a configuration of aeroelastic problems, a very large number of cases have to be run. Time accurate CFD codes are generally considered to be computationally too expensive for industrial application. Potential theory is mainly used, whereas the next level of approximation, i.e. Euler Equations with or without boundary layer coupling is only now slowly starting to find its way in the design offices despite the better approximation they provide. The application of high performance parallel computers to this kind of problems is obviously extremely interesting, not only because it allows to tackle larger problems in a shorter time but also because it opens the possibility to perform parametric studies in a reasonable time.

A time-accurate Euler code has been prepared to calculate inviscid transonic flow around oscillating 3-D wings. The code has been implemented in the NWT (Numerical Wind Tunnel) parallel supercomputer of the National Aerospace Laboratory in Japan. The objective of the present work has been to study the influence on the unsteady results of the

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

different parameters that control the calculation.

The following presents a brief description of the scheme and its parallel implementation, together with some results.

2. NUMERICAL SCHEME

Among the different schemes which have been developed to solve the unsteady 3-D Euler equations [1-10], the very popular one of Jameson [10] has been selected for this study. In the following a brief description of the implementation made here is given. More details can be found in [11].

2.1 Governing Equations

The flow is assumed to be governed by the three-dimensional time-dependent Euler equations, which for a moving domain Ω with boundary Σ may be written in integral form as:

$$\frac{\partial}{\partial t}\left(\iiint_{\Omega} U\cdot d\Omega\right)+\iint_{\Sigma}\left[(F,G,H)-\vec{V}_{\Sigma}\cdot U\right]\vec{n}\cdot dS=0 \tag{1}$$

where U is the vector of conservative flow variables; (F, G, H) are the three components of the Euler flux vector; v_Σ is the velocity of the moving boundary; and $\underline{n}$ is the unit exterior normal vector to the domain.

$$U=\begin{Bmatrix}\varrho\\ \varrho u\\ \varrho v\\ \varrho w\\ \varrho E\end{Bmatrix}\qquad F=\begin{Bmatrix}\varrho u\\ \varrho u^2+p\\ \varrho vu\\ \varrho wu\\ \varrho Eu+pu\end{Bmatrix} \tag{2}$$

$$G=\begin{Bmatrix}\varrho v\\ \varrho uv\\ \varrho v^2+p\\ \varrho wv\\ \varrho Ev+pv\end{Bmatrix}\qquad H=\begin{Bmatrix}\varrho w\\ \varrho uw\\ \varrho vw\\ \varrho w^2+p\\ \varrho Ew+pw\end{Bmatrix}$$

Here ρ, p, (u, v, w) and E respectively denote the density, pressure, cartesian velocity components of the flow, and specific total energy.

In order to close the system of equations (1) a sixth equation is needed which is obtained from the thermodynamic relationships for a perfect gas

$$p=(\gamma-1)\rho\left[E-\frac{1}{2}(u^2+v^2+w^2)\right] \tag{3}$$

2.2 Spatial Discretization

The domain around the wing is divided into an O-H mesh of hexahedral cells, for which the body-fitted curvilinear coordinates ξ,η,ζ respectively wrap around the wing profile (clockwise), normal and away from it, and along the span. Figure 1 shows an example. Individual cells are denoted by the subscripts i,j,k respectively corresponding to each of the axis in the transformed plane ξ,η,ζ.

The integral equation (1) is applied separately to each cell. Assuming that: the independent variables are known at the center of each cell; calculating the flux vector as the average of the values in the cells on either side of the face; and taking the mesh velocities as the average of the velocities of the four nodes defining the corresponding face, the following system of ordinary differential equations (one per cell) results:

$$\frac{d}{dt}(\Omega_{i,j,k}U_{i,j,k})+(Q_{i,j,k}-D_{i,j,k})=0 \tag{4}$$

where the convective operator $Q_{i,j,k}$

$$Q_{i,j,k}=\sum_{n=1}^{6}\left[(F_n,G_n,H_n)-\vec{V}_{\Sigma_n}\cdot U_n\right]\cdot\vec{S}_n \tag{5}$$

is a function of $U_{i,j,k}$, $U_{i+1,j,k}$, $U_{i-1,j,k}$, $U_{i,j+1,k}$, $U_{i,j-1,k}$, $U_{i,j,k+1}$ and $U_{i,j,k-1}$. Schemes constructed in this manner reduce to central difference schemes on cartesian meshes, and are second order accurate if the mesh is sufficiently smooth.

This formulation is inherently non-dissipative (ignoring the effect of numerical boundary conditions), so that dissipative fluxes $D_{i,j,k}$ have been added

$$D_{i,j,k}=d_{i+\frac{1}{2},j,k}-d_{i-\frac{1}{2},j,k}+d_{i,j+\frac{1}{2},k}-$$
$$-d_{i,j-\frac{1}{2},k}+d_{i,j,k+\frac{1}{2}}-d_{i,j,k-\frac{1}{2}} \tag{6}$$

The well known model of Jameson [12] is used. The idea of this adaptive scheme is to add 4th order viscous terms throughout the domain to provide a base level of dissipation sufficient to prevent non-linear instabilities, but not sufficient to prevent oscillations in the neighborhood of shock waves. In order to capture shock waves additional 2nd order viscosity terms are added locally by a sensor designed to detect discontinuities in pressure. To avoid overshoots near the shock waves produced by the combined presence of the 2nd and 4th order terms, the latter are cut off in that area by an appropriate switch.

For the dissipative flux across the face separating cells i,j,k and i+1,j,k we have (for the other faces similar expressions apply):

$$d_{i+\frac{1}{2},j,k}=\varepsilon^{(2)}_{i+\frac{1}{2},j,k}L_{i+\frac{1}{2},j,k}(U_{i+1,j,k}-U_{i-1,j,k})-$$
$$-\varepsilon^{(4)}_{i+\frac{1}{2},j,k}L_{i+\frac{1}{2},j,k}(U_{i+2,j,k}-3U_{i+1,j,k}+$$
$$+3U_{i,j,k}-U_{i-1,j,k}) \tag{7}$$

The dissipation coefficient $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$ are calculated as

$$\varepsilon^{(2)}_{i+\frac{1}{2},j,k} = k^{(2)} \max(\upsilon_{i+2,j,k}, \upsilon_{i+1,j,k}, \quad (8)$$
$$\upsilon_{i,j,k}, \upsilon_{i-1,j,k})$$

$$\varepsilon^{(4)}_{i+\frac{1}{2},j,k} = \max(0\ ,\ k^{(4)} - \varepsilon^{(2)}_{i+\frac{1}{2},j,k}) \quad (9)$$

where $L_{1+½,j}$ is a scaling factor which depends on the spectral radius of the flux Jacobian matrix in ξ direction $\lambda_{i,j,k}$

$$L_{i+\frac{1}{2},j,k} = \frac{\lambda_{i,j,k}+\lambda_{i+1,j,k}}{2} \quad (10)$$

and with

$$\upsilon_{i,j,k} = \left|\frac{p_{i+1,j,k}-2p_{i,j,k}+p_{i-1,j,k}}{p_{i+1,j,k}+2p_{i,j,k}+p_{i-1,j,k}}\right| \quad (11)$$

as a sensor of the presence of a shock wave.

2.3 Time Integration

The system of ODEs in (4) is solved by means of an explicit 5 stage Runge-Kutta scheme with two evaluations of the dissipation terms.

$$\Omega^{(0)}_{i,j,k} U^{(0)}_{i,j,k} = \Omega^{(n)}_{i,j,k} U^{(n)}_{i,j,k}$$

$$\Omega^{(1)}_{i,j,k} U^{(1)}_{i,j,k} = \Omega^{(n)}_{i,j,k} U^{(n)}_{i,j,k} - \frac{\Delta t}{4} [Q^{(0)}_{i,j,k} - D^{(0)}_{i,j,k}]$$

$$\Omega^{(2)}_{i,j,k} U^{(2)}_{i,j,k} = \Omega^{(n)}_{i,j,k} U^{(n)}_{i,j,k} - \frac{\Delta t}{6} [Q^{(1)}_{i,j,k} - D^{(1)}_{i,j,k}]$$

$$\Omega^{(3)}_{i,j,k} U^{(3)}_{i,j,k} = \Omega^{(n)}_{i,j,k} U^{(n)}_{i,j,k} - \frac{3\Delta t}{8} [Q^{(2)}_{i,j,k} - D^{(1)}_{i,j,k}]$$

$$\Omega^{(4)}_{i,j,k} U^{(4)}_{i,j,k} = \Omega^{(n)}_{i,j,k} U^{(n)}_{i,j,k} - \frac{\Delta t}{2} [Q^{(3)}_{i,j,k} - D^{(1)}_{i,j,k}]$$

$$\Omega^{(5)}_{i,j,k} U^{(5)}_{i,j,k} = \Omega^{(n)}_{i,j,k} U^{(n)}_{i,j,k} - \Delta t [Q^{(4)}_{i,j,k} - D^{(1)}_{i,j,k}]$$

$$\Omega^{(n+1)}_{i,j,k} U^{(n+1)}_{i,j,k} = \Omega^{(5)}_{i,j,k} U^{(5)}_{i,j,k}$$

$$(12)$$

which is second order accurate in time and can be shown [11] to have good diffusion and dispersion errors characteristics and less computational cost per time step than other schemes with a lesser number of stages.

2.4 Residual Averaging

This explicit time-integration scheme has a time step limit that is controlled by the size of the smallest cell.

$$\Delta t_{\max_{i,j,k}} = CFL \frac{\Omega_{i,j,k}}{\lambda_{i,j,k}+\mu_{i,j,k}+\sigma_{i,j,k}} \quad (13)$$

Even though the CFL number of the 5-stage Runge-Kutta scheme is of the order of 4, the resulting Δt's are usually too small for practical applications. This restriction can be relaxed by using a technique of residual averaging [13] which gives an implicit character to the time-integration scheme. Before each time-step the residuals $R_{i,j,k}=Q_{i,j,k}-D_{i,j,k}$ are replaced by modified residuals $R^*_{i,j,k}$ which are calculated by means of an ADI method:

$$(1-\epsilon_{i,j,k}\delta^2_\xi)(1-\epsilon_{i,j,k}\delta^2_\eta)(1-\epsilon_{i,j,k}\delta^2_\zeta) R^*_{ijk}=R_{ijk}$$

$$(14)$$

where $\delta_\xi{}^2$, $\delta_\eta{}^2$ and $\delta_\zeta{}^2$ are the second difference operators in the ξ, η, and ζ directions and $\epsilon_{i,j,k}$ is the smoothing parameter [14]

$$\epsilon_{i,j,k} = \max \{ \frac{1}{4} [(\frac{\Delta t}{\Delta t_{\max_{i,j,k}}})^2 - 1]\ ,\ 0 \}$$

$$(15)$$

with Δt denoting the desired time step.

Within a linear analysis, the former technique assures unconditional stability for any size of the time step. However, as the resulting effective Courant number becomes large the contribution of the dissipation terms to the Fourier symbol goes to zero, and consequently, the high frequencies introduced by the non-linearities are undamped [15]. Thus the practical limit for the time step is determined principally by the high frequency damping characteristics of the integration scheme used. As the properties of the 5-stage Runge-Kutta time-integration method are very good from this point of view, CFL values as high as 240 have been successfully used, which significantly decrease the calculation time needed for a typical case.

2.5 Freestream Capturing

For the scheme to satisfy the freestream capturing condition [16] it must be

$$\Omega^{(q)}_{i,j,k}=\Omega^{(n)}_{i,j,k}+\beta_q \Delta t \sum_{n=1}^{6} \vec{V}^{(q-1)}_{\Sigma n} \cdot \vec{S}^{(q-1)}_n \quad (16)$$

which is the discrete form (consistent with the numerical scheme here employed) of the Geometric Conservation Law as formulated by Thomas and Lombard [17]. It states that the cell volumes must be advanced in time in the same way as the fluid variables (even if they could be calculated analytically at each time step) to prevent grid-motion-induced errors in the numerical solution.

2.6 Boundary Conditions

The following Boundary conditions are imposed:

a) Kinematic boundary condition on the

wing surface. The pressure on the surface is extrapolated from the internal points.

b) Symmetry condition at the k=1 plane.

c) Far field boundary condition in terms of Rieman Invariants for a one dimensional flow normal to the outer computational boundary.

3. RESULTS

In the following, results for the LANN wing are presented. This is a high aspect ratio (AR=7.92) transport type wing with a 25° quarter-chord sweep angle, a taper ratio of 0.4, and a variable 12% supercritical airfoil section twisted from about 2.6° at the root to about -2.0° at the tip. The geometry used for the computational model is that of [18]. The results presented here correspond to the design cruise condition: $M_\infty=0.82$, $\alpha_0=0.6°$. The wing performs harmonical pitching oscillations about an axis at 62% root chord with an amplitude of $\alpha_1=0.25°$ and a reduced frequency k=0.104.

The calculation proceeds as follows: first an initial steady solution is obtained and quality controlled; then the time-accurate calculation is started and is time-marched until the initial transitories are damped out and an harmonic solution is obtained (typically three cycles of oscillation are needed); finally the results of the last cycle are Fourier analyzed to extract the mean value and harmonics of the different aerodynamic coefficients.

Because of the large memory and CPU time requirements of this type of methods, very few studies are available in the literature that assess the relative influence on the unsteady results of the different parameters that control the calculation. To take advantage of the benefits of parallelization to perform this task was one of the main objectives of the present work.

3.1 Artificial Viscosity

Calculations have been done for a 80x16x30 grid with different amounts of artificial viscosity, which has been varied by means of the two coefficients $k^{(2)}$ and $k^{(4)}$ in (8) and (9). Results in terms of mean part and real and imaginary parts of the first harmonic of the pressure distributions around wing sections at 17.5% and 82.5% semi-span are presented in Figures 2 to 5. Logically the main effect is on the shock resolution which in turn influences the magnitude and positions of the corresponding peaks in the first harmonic component.

3.2 Grid Density

Two different grids, namely 80x16x30 and 160x32x30, have been considered. The spanwise grid distribution and outer boundary location was kept the same for both cases, with 20 grid planes on the wing and 10 grid planes between the wing tip and the side boundary of the computational domain which is located at two semi-spans from the plane of symmetry. The outer boundary around the root section is at 9 chords. Results are shown in Figures 6 and 7, where the first harmonic of the pressure distributions around wing sections at 17.5% and 82.5% semi-span is presented. It can be seen that the influence is large as a consequence of the better shock resolution of the finer grid. On the other hand, as was to be expected, the discrepancies are much smaller when integrated along the chord to obtain sectional forces and moments, as can be seen in Figure 8 for the lift coefficient.

3.3 Time Step Size

Figures 9 and 10 respectively show the real and imaginary parts of the first harmonic of the pressure distribution around the wing section at 92.5% semi-span calculated with the 80x16x30 grid using dimensionless time-step sizes Δt^* ranging from 0.002 to 0.01 (which correspond to CFLs from 30 to 150). This section at the wing tip has been selected because at its trailing edge the smallest cells are to be found, for which the stability limit should first be reached in accordance with (13). This is indeed the case as can be clearly seen in the zoomed region. Outside of this area the results are time-step independent. Fortunately this instability is very well behaved, growing only at a very slow rate at the same time that it spreads inboard and towards the trailing edge, so that meaningful engineering calculations could be performed at larger Δt^* without a significant loss of accuracy.

3.4 Deforming vs. Rigid Moving Grids

In the present method the instantaneous grid is computed by deformation of an initial steady mesh in such a way that the grid points near the wing surface are forced to closely follow the wing (which motion is known as a function of time) whereas the displacements of grid points far from the wing surface gradually decrease and vanish at the outer boundary.

Dynamic grid deformation such as this is computationally expensive as it involves re-calculation of grid position, kinematics and metrics at each time step. For those cases in which the wing has no elastic deformations it is also possible to perform the calculation on a grid that moves with the wing as a rigid body. This option, although theoretically less accurate than the former, is obviously computationally less expensive. To evaluate its influence on the results, calculations have been performed both with the usual deforming grid and with a rigid one. It has been found that differences are negligible. No figure is given because the differences are within the resolution of the graph.

3.5 Freestream Capturing

Calculations have been performed both imposing and not imposing the freestream capturing condition (16). In the latter case the cell volumes at each time step have been calculated analytically. Again the differences in the results are totally negligible.

4. PARALLEL IMPLEMENTATION IN NWT

The above presented scheme was originally developed in a Cray-YMP computer and has been implemented in the NWT (Numerical Wind Tunnel) machine of the National Aerospace Laboratory [19]. This is a distributed memory parallel machine with 140 vector processing elements (PE) and two Control Processors connected by a cross-bar network.

Each PE is itself a vector supercomputer similar to Fujitsu VP400 with a peak performance of 1.7 GFlops and includes: 256 Mbytes of main memory, a vector unit, a scalar unit and a data mover which communicates with other PE's. The resulting total performance of NWT is 236 GFlops and 35 GBytes.

The code has been parallelized using Fujitsu NWT FORTRAN which is a FORTRAN 77 extension to perform efficiently on distributed memory type parallel computers. The extension is realized by compiler directives. Basic execution method is the spread/barrier method.

The present scheme has always two directions in which the computation can be performed simultaneously. Accordingly we can use one direction for vectorization and the other for parallelization. For the O-H grid used here the most natural way of parallelizing, i.e. assigning different vertical grid planes to different processing elements has been used. We thus divide every array evenly along the k-index and assign each part to different PEs. The vectorization is made in i-direction which usually has the largest number of cells.

With this partition, i-derivatives and j-derivatives can be computed in each PE without any communication. The computation of k-derivatives in PE_k requires data stored in PE_{k+1} and PE_{k-1} which, in principle, would imply the need to communicate with the neighbor PEs, thus increasing the overhead. This is avoided using overlapped partitioned arrays. Array partitions are defined in such a way that adjacent partitioned ranges automatically overlap and have some common indices (with a depth depending on the stencil) so that copies of selected data at the interfaces between two PEs are stored at both local memories. In this way k-derivatives can also be computed in each PE without any communication. At the end of each calculation cycle, data in the overlap range of the partitioned arrays is harmonized by copying its value from the parent PE.

The above explained procedure can be maintained throughout the code except at the residual averaging subroutine, where the alternating directions method (ADI) employed prevents its use as it requires a sequential calculation. The inversions in the i- and j-directions can be done in each PE independently so that the k-parallelization can be maintained, with the vectorization in j-direction for the i-inversion and in i-direction for the j-inversion. As for the k-inversion, the process must be sequential in the k-direction so that we transfer the affected data from a k-partition to a j-partition. Then we can compute the k-inversion on each PE with vectorization in i-direction. At the end of the calculation the data is transferred back to a k-partition. Figure 11 depicts the calculation flow.

In Figure 12 the speed-up factor (ratio of CPU time in 1 PE to CPU time in n PEs) vs. number of PEs used is presented for calculations performed for the LANN wing with the 160x32x30 grid. The results strongly depend on whether the residual averaging technique is used or not, because of the need to transfer data between partitions. Its relative importance in relation to the normal data transfer workload decreases as the number of PEs used increases and both curves tend to reach a common limit. It must be born in mind that the 160x32x30 grid only fills about 20% of the main memory of a single PE (less than 1% when 32 are used), so that the granularity of the problem is extremely low. The parallel efficiency is expected to dramatically increase for larger grids, as has been the case with other codes [20].

An indication of the CPU times required to march in time the solution for one period of oscillation (using a Δt^* of 0.01 for the coarse grid and 0.004 for the fine one which respectively correspond to CFLs of 150 and 240) is given in Table 1.

5. CONCLUDING REMARKS

A time-accurate Euler code to calculate unsteady transonic flow about oscillating wings has been prepared and implemented in the NWT parallel supercomputer. The achieved performance has shown the feasibility of using this type of computationally expensive methods in an engineering environment. The influence of different parameters on unsteady computations has been studied.

ACKNOWLEDGEMENTS

This work was supported by a fellowship from the SCIENCE AND TECHNOLOGY AGENCY of Japan.

REFERENCES

[1] Whitfield, D.L.; Janus, J.M.; "Three-Dimensional Unsteady Euler Equations Solution Using Flux Vector Splitting"; AIAA Paper 84-1552; 1984

[2] Salmond, D.J.; "Calculation of Harmonic Aerodynamic Forces on Airfoils and Wings from the Euler Equations"; RAE Tech. Memo Aero 2011,; 1984

[3] Sankar, L.N.; Wake, B.E.; Lekoudis, S.G.; "Solution of the Unsteady Euler Equations for Fixed and Rotor Wing Configurations"; Journal of Aircraft Vol. 23, No. 4, pp. 283-289; 1986

[4] Sankar, L.N.; Malone, J.B.; Schuster, D.; "Euler Solutions for Transonic Flow Past a Fighter Wing"; Journal of Aircraft, Vol. 24, No. 1, pp.10-16; 1987

[5] Belk, D.M.; Whitfield, D.L.; "Time-Accurate Euler Equations Solutions on Dynamic Blocked Grids"; AIAA Paper 87-1127, 1987

[6] Anderson, W. K.; Thomas, J.L.; Rumsey, C.L.; "Extension and Applications of Flux-Vector Splitting to Unsteady Calculations on Dynamic Meshes"; AIAA Paper 87-1152, 1987

[7] Brenneis, A.; Eberle, A.; "Application of an Implicit Relaxation Method Solving the Euler Equations for Time-Accurate Unsteady Problems"; J. of Fluid Engineering, Vol. 112, pp. 510-520; 1990

[8] Guruswamy, G.P.; "Unsteady Aerodynamic and Aeroelastic Calculations for Wings Using Euler Equations"; AIAA J., Vol. 28, No. 3, pp. 461-469; 1990

[9] Brenneis, A.; Eberle, A.; "Evaluation of an Unsteady Implicit Euler Code Against Two and Three-Dimensional Standard Configurations"; AGARD CP-507, Paper 10,; 1992

[10] Jameson, A.; Venkatakrishnan, V.; "Transonic Flows about Oscillating Airfoils using the Euler Equations"; AIAA Paper 85-1514, 1985

[11] Ruiz-Calavera; "Calculation of Unsteady Transonic Aerodynamic Loads on Wings Using the Euler Equations"; INTA OAT/TNO/4510/005/INTA/95, 1995

[12] Jameson, A.; "A non-oscillatory Shock Capturing Scheme using Flux Limited Dissipation". Princeton University MAE Report 1653, 1984

[13] Jameson, A.; "Transonic Flow Calculations for Aircraft"; Lecture Notes in Mathematics, Vol. 1127; Numerical Methods in Fluid Dynamics; Editor: F. Brezzi; Springer Verlag; pp. 156-242; 1985

[14] Batina, J.; "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes"; AIAA J., Vol. 28, No. 8, pp.1381-1388; 1990

[15] Radespiel, R.; Rossow, C.; "An Efficient Cell-Vertex Multigrid Scheme for the Three-Dimensional Navier-Stokes Equations"; AIAA Paper 89-1953; 1989

[16] Obayashi, S.; "Freestream Capturing for Moving Coordinates in Three Dimensions"; AIAA Journal, Vol. 30, No. 4, 1991

[17] Thomas, P.D.; Lombard, C.K.; "Geometric Conservation Law and its Application to Flow Computation on Moving Grids"; AIAA Journal, Vol. 17, No. 10, pp. 1030-1037, 1979

[18] "AGARD Three-Dimensional Aeroelastic Configurations"; AGARD-AR-167; 1982

[19] Hirose, N.; "Numerical Wind Tunnel Project and Computational Fluid Dynamics at National Aerospace Laboratory, Japan"; NAL TM-648T

[20] Miyoshi, H. et al.; "Development and Achievement of NAL Numerical Wind Tunnel (NWT) for CFD Computations"; Proceedings of IEEE Super Computing 1994

Table 1. Computational Performance

	NWT 1 PE	NWT 32 PE	CRAY-YMP (M92) 1 PE
80x16x30	15'	2.5'	59'
160x32x30	187'	31'	-

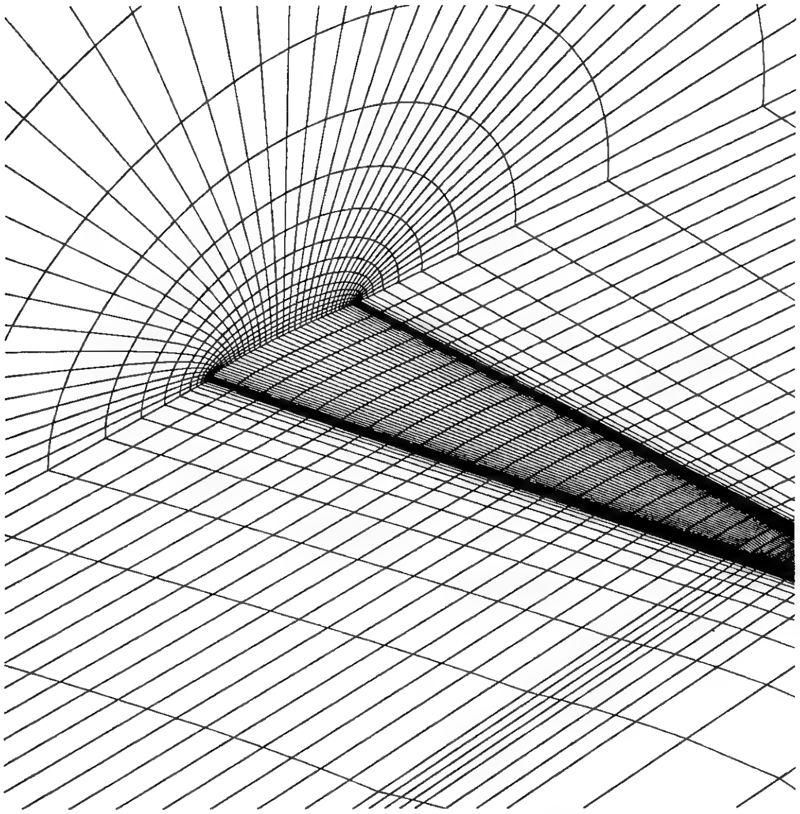


Fig. 1. LANN Wing. 80x16x30 Grid.

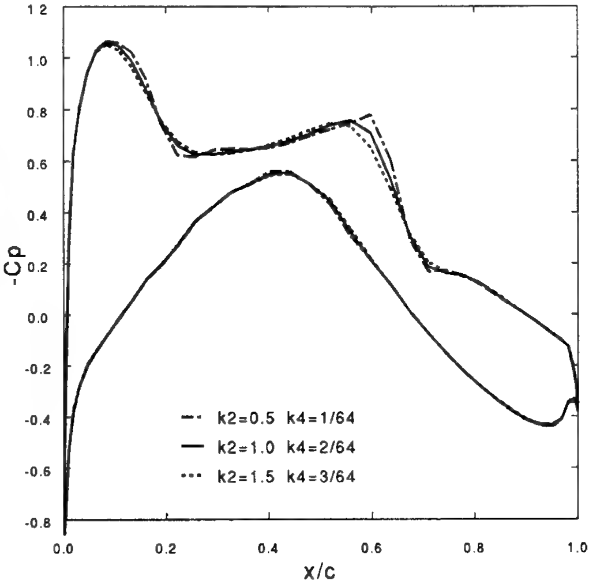


Fig. 2: Mean Part. 17.5% semispan

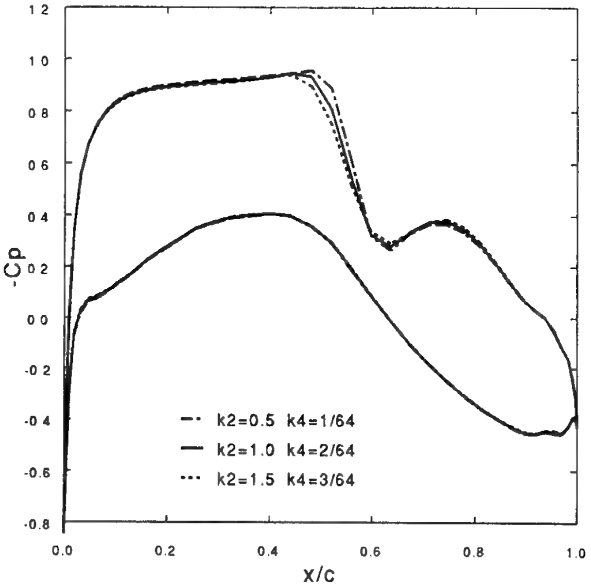


Fig. 3: Mean Part. 82.5% semispan

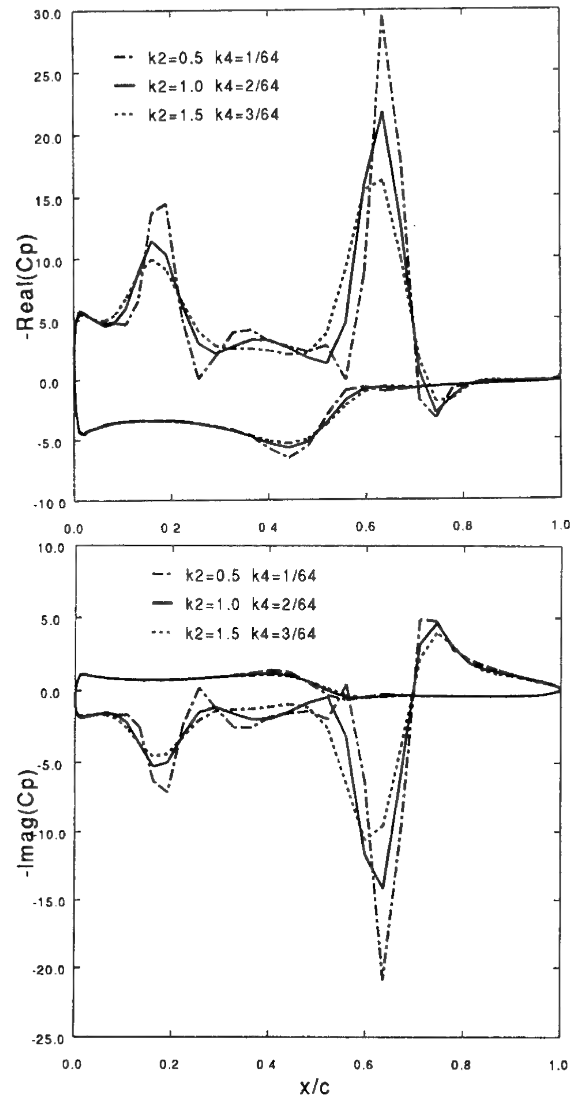


Fig. 4: First Harmonic. 17.5% semispan

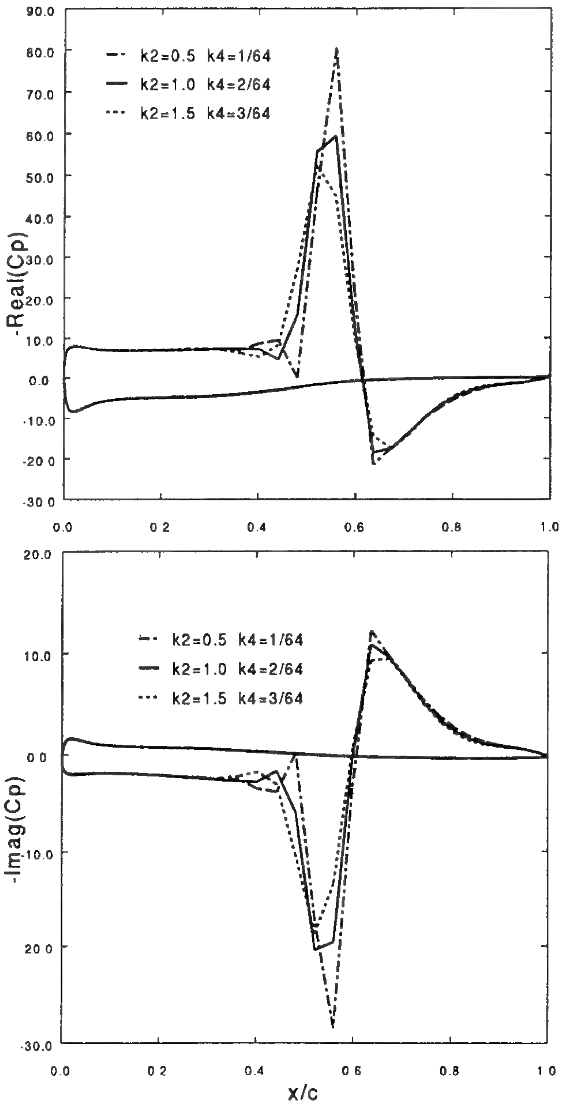


Fig. 5: First Harmonic. 82.5% semispan

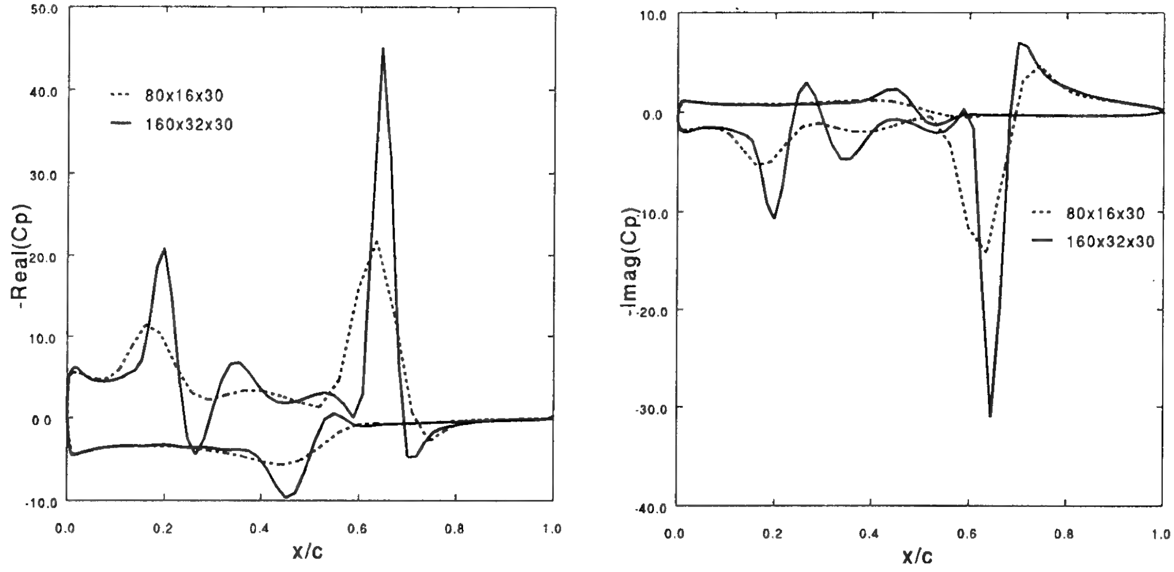


Fig. 6: First Harmonic. 17.5% semispan

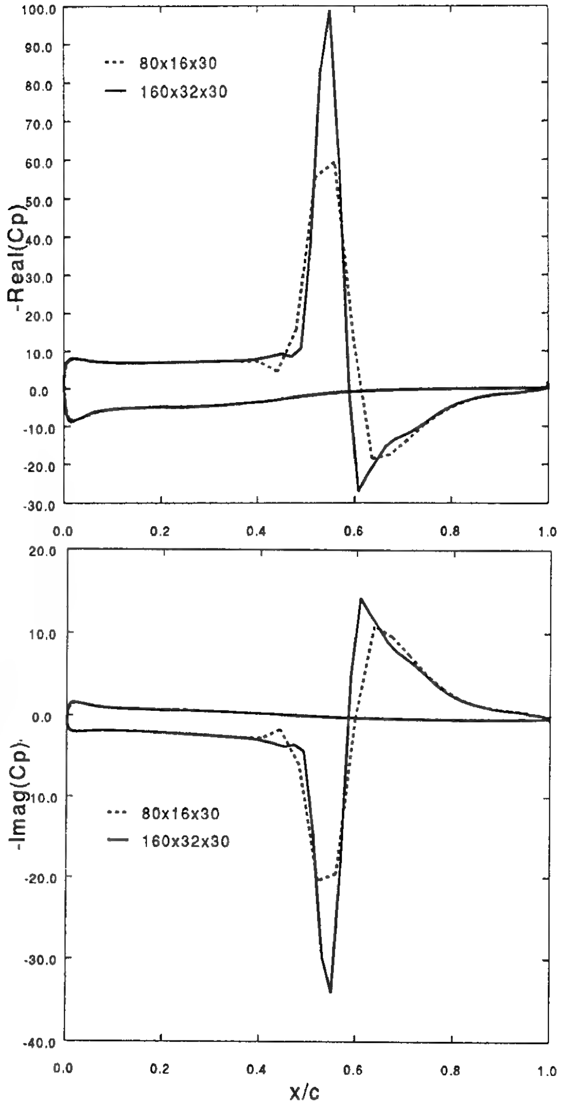


Fig. 7: First Harmonic. 82.5% semispan

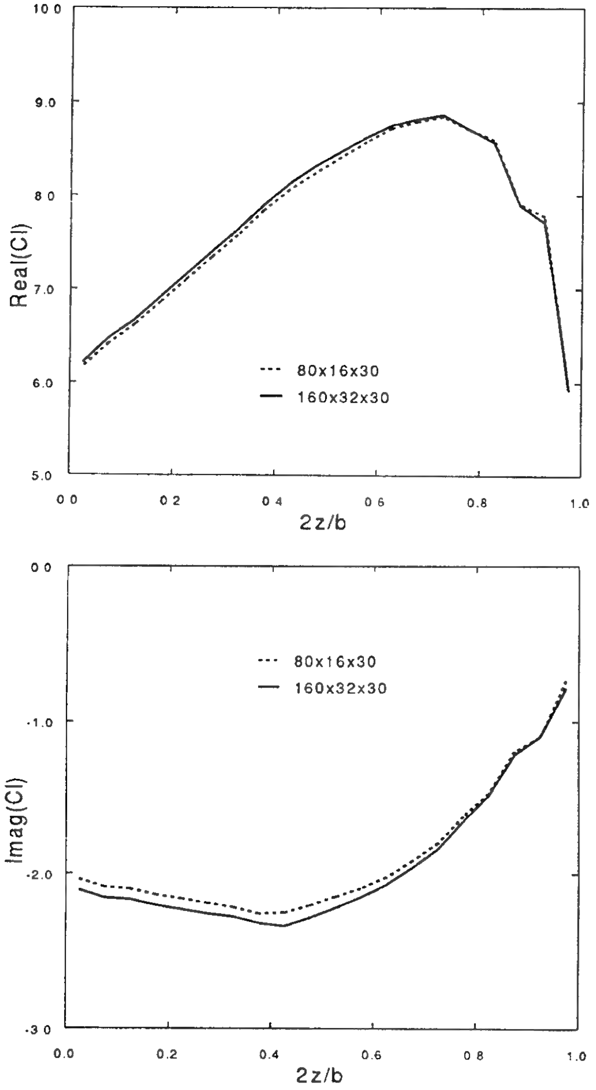


Fig. 8: Lift Coefficient. 1st Harmonic.

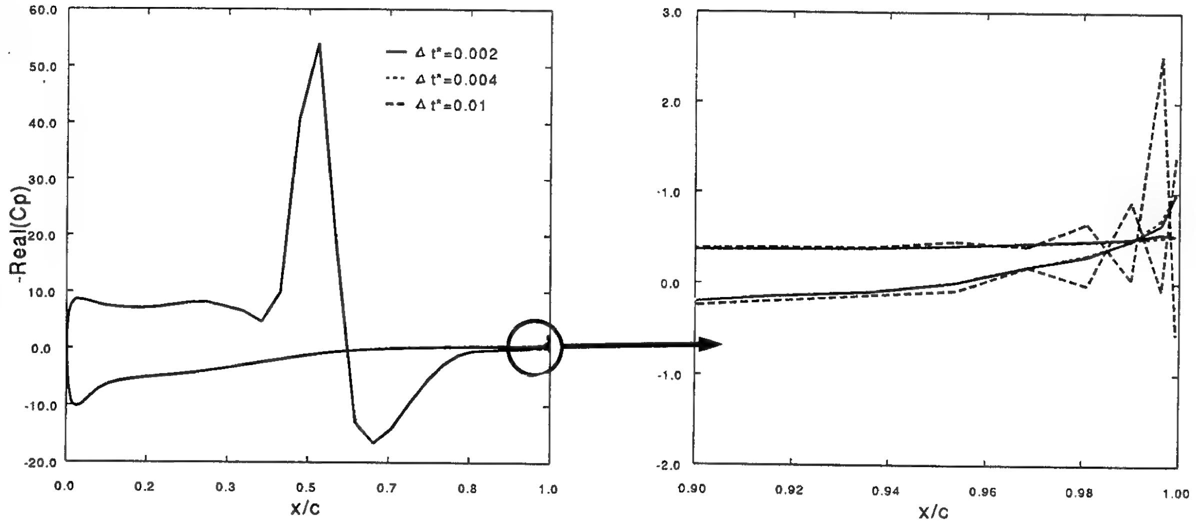


Fig. 9: Real Part. First Harmonic. 92.5%

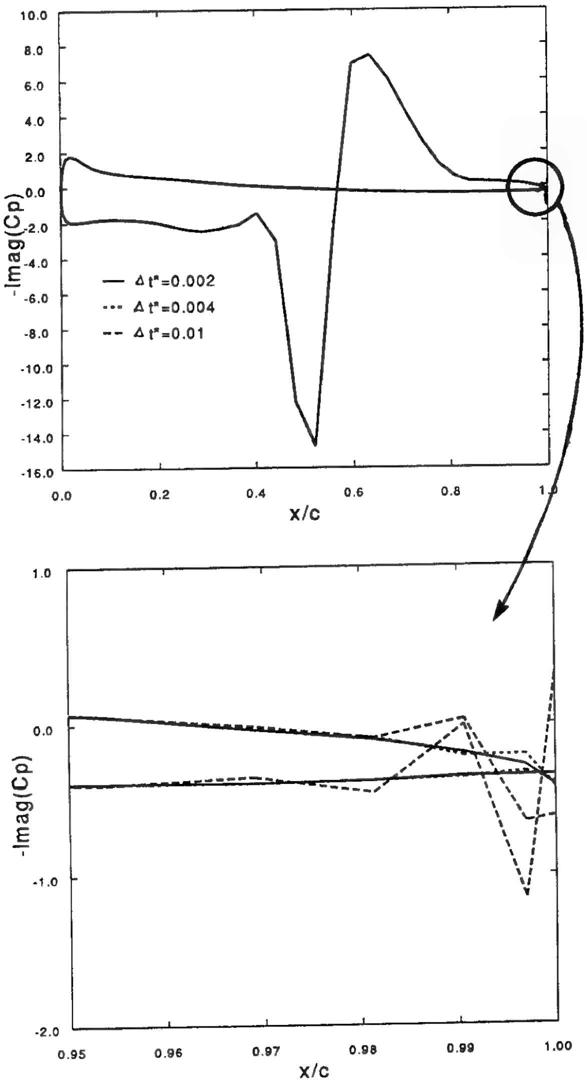


Fig. 10: Imaginary Part.
First Harmonic. 92.5%

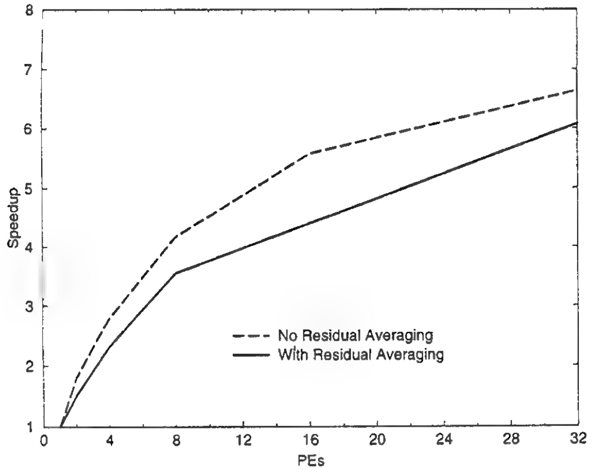


Fig. 12: Speed-up Factor.

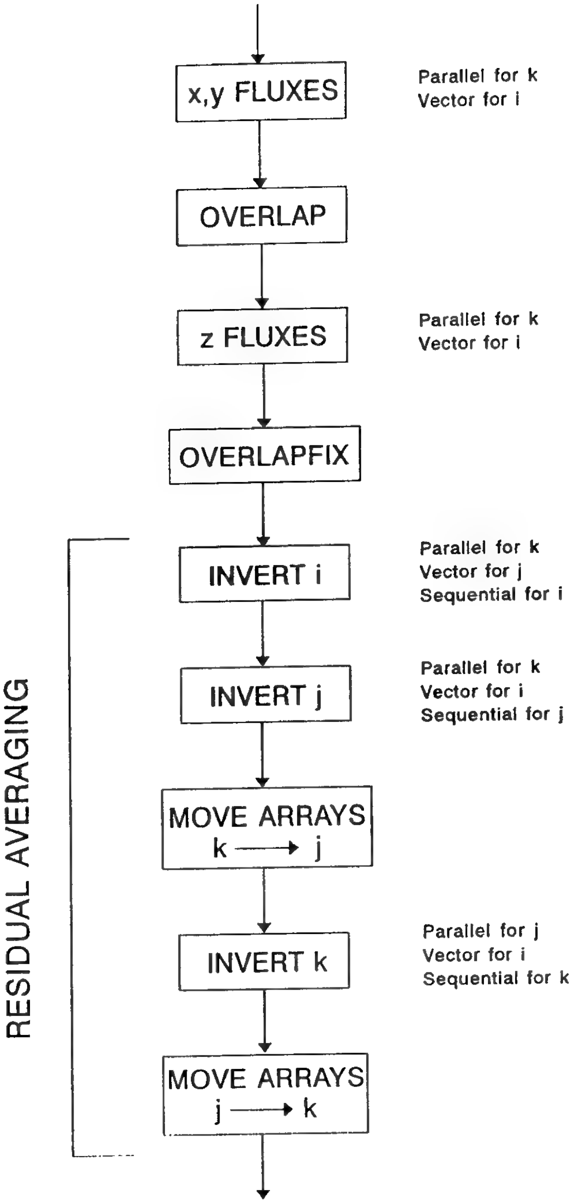


Fig. 11: Calculation Flow.

Parallel Implicit Upwind Methods for the Aerodynamics of Aerospace Vehicles

K.J. Badcock[1] and B.E. Richards [2]

Abstract. Research at the University of Glasgow, based around implicit methods for solving the Euler and Reynolds' Averaged Navier-Stokes equations and to be reported in this paper, has targeted advanced CFD methods for tackling the complex flow fields of interest to aerospace vehicle designers. The requirements for this application are for efficient, high resolution schemes which can be ported to various MPP systems and implemented with robustness to give fast turn round times at competitive cost. It is recognised that the most demanding topics concern unsteady viscous flows and thus time accuracy and efficiency is pursued as a high priority. This paper then reviews the work, ongoing and planned, by the team at Glasgow in code developments embracing future computing environments and including some results not previously published. The example test cases used in the performance and sensitivity studies include the transonic flow results on the RAE 2822 aerofoil and ONERA M6 wing selected by AGARD. The computing environments to which the codes port include workstations, either used singly or clustered to provide a parallel computing domain, and also integrated distributed memory Supercomputers such as CRAY T3D and Intel Hypercube systems. The paper outlines these technologies also.

1 Introduction

Aerodynamics has been established as a foundation technology for the design of aerospace vehicles. Good application of aerodynamics will lead to substantial economic benefits for future aircraft designs. Particularly important target areas include drag reduction to improve direct operating costs and better prediction of steady and unsteady loads on aircraft to overcome structural conservatism at the time of freezing the design. For the majority of aircraft, this requires the particular capability of predicting the phenomena of shock waves and flow separation. This can be achieved through a better understanding of the fluid mechanics of flow interactions using either experimental techniques or computational methods. Wind tunnel testing at simulated conditions, particularly for flutter, for example, is becoming increasingly expensive. On the other hand, with the rapid developments in computer hardware and computational techniques, the topic of computational fluid dynamics is reaching maturity as a viable way of providing design solutions.

A reasonable simulation of the fluid dynamics of high Reynolds' number can be obtained by solving the Reynolds' averaged Navier-Stokes (RANS) equations. Increasing computer power now makes the solution of these equations feasible. The level of turbulence model used needs to be a compromise between a simple eddy viscosity model such as Baldwin Lomax and a more complex second moment closure model. In this work the former is used, but the codes are starting to use the more general k-ω two equation model.

To satisfy the general requirements for a code suited for aircraft design, it should be accurate, efficient and robust and usable on future computer architectures. The general approach chosen by the University of Glasgow CFD Team in this work is to use high order upwind differencing to provide accuracy and robustness and to mostly use implicit methods to provide efficiency [5] [9]. Unstructured grids are also being considered by the Team as a way forward for dealing with geometric complexity but there are developmental difficulties in tackling viscous flows near boundaries and calls for high memory. Geometric complexity using structured meshes can be accommodated using multi-block grids which lend themselves to distributed memory computing architectures using a multi-domain approach. The combination of an implicit approach on a structured grid for wall turbulent flows provides an efficient code, particularly for unsteady flows.

There exists a considerable variety of computer architectures from which to choose. The general consensus, however, is that competitively priced distributed memory massively parallel processors (MPPs) will provide the Teraflops facility (or greater) that will be required to tackle CFD solutions using the RANS model for flows over complete aircraft configurations. A number of vendors promise production of such Teraflops facilities in the near future, although the cost is likely to be beyond the means of all but the largest organisations. Also there needs to be a further investment in adapting the majority of existing codes to use it. There is a trend to provide a similar architecture at a much lower cost using workstation clusters.

[1] Lecturer, Department of Aerospace Engineering, University of Glasgow, Glasgow, G12 8QQ, UK

[2] Professor, Department of Aerospace Engineering, University of Glasgow, Glasgow, G12 8QQ, UK

Paper presented at the AGARD FDP Symposium on "Progress and Challenges in CFD Methods and Algorithms" held in Seville, Spain, from 2-5 October 1995, and published in CP-578.

On the sort of broad-bandwidth networks that are planned for the future in Corporate networks, the type of powerful high memory workstations that are being used for detailed CAD/CAM design work in the engineering industry in the daytime are amenable to be turned loose at off peak times to provide a powerful high-memory system.

Creating the parallel computing environment for the Glasgow CFD Team has proven to be an interesting case history that it is appropriate to relate as a contribution towards the theme of this conference. Before 1990, computer systems within the Universities in the UK had undergone a major upgrade each seven years, funded by the University Funding Council, and this generally enabled the acquisition of a useful multi-user mainframe. Numerically intensive computer users then had access to off peak cycles through batch facilities. Before 1994 at Glasgow, for example, the sizeable University central Computing Service operated a CMS environment on an IBM 3090 150E vector facility for scientific work, with the help of a technological agreement with the vendor, as well as VME and VMS environments on sizeable ICL and DEC facilities, respectively. From another initiative the University also acquired a 32 transputer distributed memory Meiko Computing Surface, along with a systems manager, on which some early experience on parallel computing was developed by the CFD Team members to complement time awarded by peer reviewed on National Facilities such as CRAY-XMP and YMP vector multi-processors . The Team's work could be classed at this stage in the category of high performance computing (HPC).

In 1994, the Funding Council support changed to a system of IT support on an annual basis, at the same time the University adopted an IT strategy to distributed the monies involved thinly to all Departments whilst providing a core support for: the overall Campus Network including a FDDI backbone (later an ATM backbone); and a UNIX cluster for core computing (with a cost imposed on groups who used cycles above a threshold which was set at a low level). The implication of this University strategy was the need for HPC users to prise a proportion of their Department's allocation of funds and add it to other initiatives to secure the computing environment that they needed. Also at about the same time at National level, resources targeted for research (managed by EPSRC) were used to purchase a CRAY T3D with 320 DEC-Alpha nodes and following bids this was placed at the Edinburgh Parallel Computing Centre at Edinburgh University. This facility was designated for the exclusive use of a limited number of University Consortia to tackle Grand Challenge problems only.

With this background, the team then fronted two main initiatives to achieve an acceptable computing resource for its ambitions to develop state-of-the-art code that might be useful for aircraft designers. The first was to develop a University Consortium (finally, this included the Universities of Bristol, Glasgow, Oxford and Swansea and UMIST) that proposed a topic on *Physically and Geometrically Complex Aerodynamic Flows for Aircraft Flight* to use the Cray T3D facility. The title was changed to the shortened form *Computation of Complex Aerodynamic Flows* or CCAF Project after the proposal was accepted [2]. The other initiative was to develop a Consortium of Departments within the University to bid for resources under the New Technologies Initiative (NTI) for the development of a High Performance Parallel Computing facility from Spare Capacity on a Network of Workstations.[3] NTI was developed by the Joint Information Systems Committee from funds that the Committee had secured themselves from the Higher Education Funding Councils to promote pilot studies towards developing state-of-the-art computing capabilities across the Universities. When the funds were awarded the University project was designated the HNW Project. These projects (both now have a year's maturity) give access to a world class resource to the CFD Team. These projects will now be described separately.

One target area of application for the CCAF project is towards the study of aeroelasticity at the edges of the flight envelope, an area in which the non-linearity of the problem poses considerable uncertainties and is likely to reveal interesting new mechanics. The challenge is to be in a position to complement experimental and analytical studies of these complex physical phenomena using facilities as powerful as the EPCC Cray T3D. Electrodynamics radiation is also included in the programme because of the commonality in grids and solution techniques and the opportunity to widen the application base of the project. The resource awarded is modest (around 64,000 processor hours per year), but with development being done on local computing environments with production tests done on the National facility, the resource is useful.

Two main computational approaches are being pursued in CCAF: structured grid work is at a more mature stage, parallelisation of multi-block codes and dealing with boundary layers is straight forward but dealing with geometric complexity is problematic; unstructured grid work copes well with geometric complexity but partitioning causes problems. The project includes comparisons between codes developed in order to determine the best future strategy. In the area of aeroelasticity, there is a dearth of experimental data of the quality and appropriateness for CFD validation. Nevertheless the Consortium has identified a suitable unsteady test case involving the AGARD LANN swept wing to provide an appropriately challenging common test case. The Glasgow Team is involved particularly with the development of a multi-block structured grid flow code meshed with a structural code made available from industry and uses on average 1,600 processor hours of T3D resource per month on this. Some preliminary results are reported below.

At the other end of the cost scale, the HNW *cluster* project was awarded 8 man years of effort by JISC over a period of 3 years. The six collaborating Departments in the University of Glasgow provided funds to purchase equipment and software for a pilot facility, which could also be used as a demonstrator for a dedicated cluster as well as a base for testing different

[3] see http://www.aero.gla.ac.uk/Research/HNW for full details

cluster technologies. Following a stringent selection process and based on the company's strong interest in the cluster technology, six Silicon Graphics' Indys with MIPS R4400 processors and 64 MB memory and 17 inch monitors were selected and purchased. These were assembled together in one laboratory and connected using grade 5 UTP cabling to a 10baseTEthernet switch, which is the standard presently used by the University, and this itself was connected to the network. Using PVM 3.3 message passing, excellent performance was achieved using the Team's CFD codes, with little latency [6]. A planned upgrade to ATM switching on the UTP cabling is planned in the near future to improve communication speed as well as a multi-cluster activity with an adjoining University linked to the local ATM based Metropolitan Area Network (MAN) called ClydeNet.

From other research projects, ten more Indys have recently been added to the Departmental 10baseT network. The combined resource is available generally as a computing domain to users given an account. Apart from PVM being installed on the cluster as the message passing software for the parallel implementations, alternatives for users include MPI (CHIMP and LAM versions) as well as Oxford Parallel BSP. Clusters in other Departments in the University are beginning to be set up in a similar way.

Because of the heterogeneous nature of the user base of the cluster, a resource management system was required to optimise use of these cluster resources. The public domain software NQS, CONDOR and DQS and demonstration versions of the supported software CODINE^{TM} and LSF^{TM} were obtained and assessed. LSF^{TM}, written by Platform Computing Inc. of Toronto, had the best ingredients for the University based project, particularly a multi-cluster capability, and has been selected by a number of Industries, particularly some Aerospace Industries as a means of managing the cluster resource. A University agreement, which included technological exchanges towards the future development of LSF^{TM}, made available a multi-platform site license to explore its use in a University environment and particularly this presently unique facility of managing multi-clusters. The experience to date in its implementation is that improved load balancing, and hence a considerably better use of cycles is made by now submitting jobs to the domain, rather than to a specific workstation. The software identifies the best resource for a job and carries it out transparently to the user. If a user wishes to reclaim use of a machine for interactive work, the part of the job being done on that machine is automatically checkpointed and migrated to another machine with spare capacity. PVM is embedded in the software so that it provides an ideal system for queuing and implementing parallel programmes at low cost. LSF^{TM} provides excellent user interfaces, which help system managers of clusters to improve their service to users.

With continued development of the cluster technology, there is evidence that this type of *affordable computing* could be a norm in design offices within the Aerospace Industry. With this background on the technology used at Glasgow, Sections 2 and 3 of the paper, outline the discretisation methodologies for the two and three dimensional codes and provides some new examples and Section 4 discusses the parallel coding methodology used.

2 Two-Dimensional Method

The two dimensional thin-layer Reynolds' Averaged Navier-Stokes equations in generalised curvilinear co-ordinates (ξ,η) with η normal to the surface can be denoted in non-dimensional conservative form by

$$\frac{\partial \mathbf{w}}{\partial t}+\frac{\partial \mathbf{f}}{\partial \xi}+\frac{\partial \mathbf{g}}{\partial \eta}=\frac{\partial \mathbf{s}}{\partial \eta} \qquad (1)$$

where w denotes the vector of conserved variables, f the convective streamwise flux, g the convective normal flux and s the normal viscous flux.

One implicit step, updating the primitive variables $\mathcal{P}$, can be written as

$$(\frac{\partial \mathbf{w}}{\partial \mathcal{P}}+\Delta t\frac{\partial \mathbf{R}_\xi}{\partial \mathcal{P}}^n+\Delta t\frac{\partial \mathbf{R}_\eta}{\partial \mathcal{P}}^n)\delta\mathcal{P}=-\Delta t(\mathbf{R}_\xi^n+\mathbf{R}_\eta^n) \qquad (2)$$

where $\mathbf{R}_\xi$ and $\mathbf{R}_\eta$ are terms arising from the spatial discretisation in the ξ and η directions respectively and

$$\frac{\partial \mathbf{f}}{\partial \xi}\approx \mathbf{R}_\xi \qquad \frac{\partial(\mathbf{g}-\mathbf{s})}{\partial \eta}\approx \mathbf{R}_\eta .$$

$$\delta\mathcal{P}=\mathcal{P}^{n+1}-\mathcal{P}^n$$

In the present work the spatial terms are discretised using Osher's flux approximation with MUSCL interpolation and the Von Albada limiter for the convective terms and central differencing for the viscous fluxes. The Baldwin-Lomax turbulence model is employed to provide a turbulent contribution to the viscosity but this is not linearised in time in the present work, i.e. turbulence contributions only appear on the right-hand-side of equation (2). This has been found not to degrade the stability properties of the methods examined in this paper.

The alternating direction implicit version of equation (2) is

$$(\frac{\partial \mathbf{w}}{\partial \mathcal{P}}+\Delta t\frac{\partial \mathbf{R}_\xi}{\partial \mathcal{P}}^n)(\frac{\partial \mathbf{w}}{\partial \mathcal{P}})^{-1}(\frac{\partial \mathbf{w}}{\partial \mathcal{P}}+\Delta t\frac{\partial \mathbf{R}_\eta}{\partial \mathcal{P}}^n)\delta\mathcal{P}^n=\mathbf{R}_{exp} \qquad (3)$$

where

$$\mathbf{R}_{exp}=-\Delta t(\mathbf{R}_\xi^n+\mathbf{R}_\eta^n).$$

The ADI factorisation which appears on the left hand side of equation (3) has been widely used to approximate a solution to the system (2) because the banded structure of each of the factors makes it relatively easy to solve. However, the solution of the ADI system is not an exact solution of equation (2) and in practice the factorisation error (the error introduced by solving equation (3) rather than equation (2)) leads to a practical limit on the time step and introduces another source of error into the calculation. This motivates the use of a preconditioned conjugate gradient solution of the unfactored system.

Conjugate gradient methods find an approximation to the solution of a linear system by minimising a suitable residual error function in a finite dimensional space of potential solution vectors. Several algorithms are available including BiCG, CGSTAB, CGS and GMRES. These methods were tested in [3] and it was concluded that the choice of method is not as crucial as the preconditioning. However, the CGS method was found to be the quickest of the three methods that do not require re-orthogonalisation and is used here. CGS has the additional advantage that the transpose of the matrix on the left-hand side of equation (2), is not required, reducing implementation difficulties. The CGS algorithm was derived in [10] and is restated in [12].

Denoting the linear system to be solved at each time step by

$$A\text{x}=\text{b} \tag{4}$$

we seek an approximation to $A^{-1} \approx C^{-1}$ which yields a system

$$C^{-1}A\text{x}=C^{-1}\text{b} \tag{5}$$

more amenable to conjugate gradient methods. The ADI method provides a fast way of calculating an approximate solution to equation (4) or, restating this, of forming the matrix vector product

$$C^{-1}\text{b}=\text{x}. \tag{6}$$

Hence, if we use the inverse of the ADI factorisation as the preconditioner then multiplying a vector by the preconditioner can be achieved simply by solving a linear system with the right-hand side given by the multiplicand and the left hand side matrix given the approximate factorisation. The factors in C are put in triangular form once at each time step with the row operations being stored for use at each multiplication by the preconditioner. This roughly doubles the storage requirements of the method.

To illustrate the performance of this method we present results for flow over an RAE2822 aerofoil at a free stream Mach number of 0.73, an angle of attack of 2.73^o and a Reynolds number of 6.5×10^6. The comparison of convergence rates for the present method (called AF-CGS), straight ADI and an explicit local time-stepping method was made in [5] and an improvement in time to convergence of 25 per-cent was noted for the present method when compared with ADI. When the free stream flow is used as starting conditions, the CFL number which yields fastest convergence for the AF-CGS method is 35 but CFL numbers of up to 50 can be used. The largest CFL number which yields a solution for ADI is 18 and hence, removing the factorisation error allows the use of larger time steps. A further reduction in the time to convergence by a factor of five has been achieved by mesh sequencing. Three levels of mesh sequencing were used to provide a good starting solution on the finest mesh (257x65). Using this approach the optimal CFL number was increased to 100 and the overall time to converge to within 0.25 per cent of the fully converged lift value was reduced by a factor of 5. The time to convergence as a function of CFL number is plotted in figure 1 and shows a clear minimum. This is because there is a balance between increasing the CFL number to minimise the number of implicit steps and reducing the CFL number to minimise the number of CGS steps at each implicit step. The comparison of the pressure distribution with experiment for various levels of convergence is shown in figure 2 and shows good agreement with experiment.

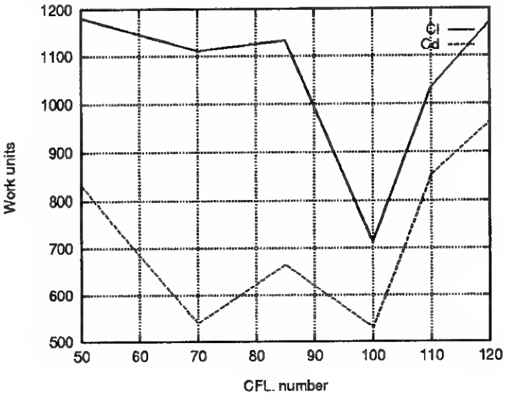


Figure 1. *Time to converge to within 0.25 % of drag as function of CFL number on finest grid*

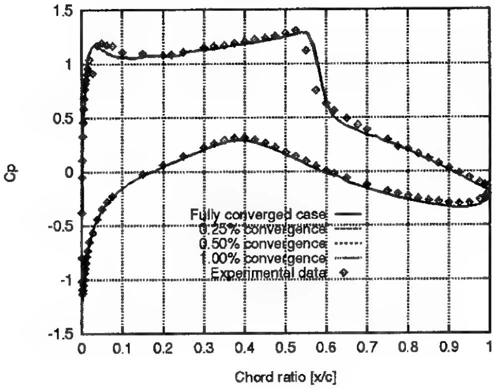


Figure 2. *pressure distribution at various levels of convergence at a CFL number of 100 on the 257x57 grid*

A similar approach has been used for unsteady flows over pitching and plunging aerofoils and aerofoils with moving flaps [4]. The main conclusion from this work was that AFCGS does not allow the choice of time step from purely accuracy considerations because of the need to limit the time step to ensure the reasonable performance of the linear solver. However, AF-CGS does allow for larger time steps and a reduced computational cost when compared with ADI. For one particular case the stability restriction on the size of the time step is a global CFL number of 1000. The average CFL number during one cycle for the unfactored method is around 2000 for the unfactored method translating into a saving in CPU time of around twenty-five per-cent.

3 Three-dimensional Extensions

The extension of the method to three-dimensions is complicated by two considerations. First, computer storage becomes a limiting factor due to the need to store large Jacobian matrices. Secondly, the ADI factorisation in three-dimensions is significantly worse than in two-dimensions, making its use as a preconditioner less favourable. This fact however means that there are increased gains to be made in three dimensions by the use of an alternative to ADI.

One step of the method considered can be written as

$$(\frac{\partial w}{\partial \mathcal{P}}+\Delta t\frac{\partial R_x}{\partial \mathcal{P}}+\Delta t\frac{\partial R_y}{\partial \mathcal{P}})\frac{\partial w}{\partial \mathcal{P}}^{-1}(\frac{\partial w}{\partial \mathcal{P}}+\Delta t\frac{\partial R_z}{\partial \mathcal{P}})\delta\mathcal{P}=R_{exp} \quad (7)$$

where

$$R_{exp}=-\Delta t(\mathbf{R}_x+\mathbf{R}_y+\mathbf{R}_z).$$

This two factor step can be loosely described as unfactored in each spanwise slice and approximately factored in the spanwise direction. A stability analysis [7] has shown that the method has similar stability properties to the two factor ADI method in two-dimensions, representing a significant improvement on the behaviour of the three factor method in three-dimensions. The linear system resulting from the first factor in equation 7 has a more complicated structure than the block pentadiagonal systems which are encountered for each factor in the three factor method. However, this sytem can be solved using a direct generalisation of the method described for two dimensions above i.e. we solve the system

$$C^{-1}Ax=C^{-1}b \quad (8)$$

by the CGS method where

$$A=(\frac{\partial w}{\partial \mathcal{P}}+\Delta t\frac{\partial R_x}{\partial \mathcal{P}}+\Delta t\frac{\partial R_y}{\partial \mathcal{P}}), \quad (9)$$

$$C=(\frac{\partial w}{\partial \mathcal{P}}+\Delta t\frac{\partial R_x}{\partial \mathcal{P}})\frac{\partial w}{\partial \mathcal{P}}^{-1}(\frac{\partial w}{\partial \mathcal{P}}+\Delta t\frac{\partial R_y}{\partial \mathcal{P}}) \quad (10)$$

and

$$b=-\Delta t(\mathbf{R}_x+\mathbf{R}_y+\mathbf{R}_z), \quad (11)$$

followed by the solution of a block pentadiagonal system for the updates

$$(\frac{\partial w}{\partial \mathcal{P}}+\Delta t\frac{\partial R_z}{\partial \mathcal{P}})\delta\mathcal{P}=x. \quad (12)$$

The two factor method has substantially reduced memory requirements compared with the fully unfactored method. For the third order spatial discretisation there are 13 non-zero 5 by 5 blocks for the rows in the unfactored matrix associated with any one grid cell. This means that the number of floating point numbers which must be stored for the coefficient matrix for a mesh with $\mathcal{N}$ cells is $325\mathcal{N}$. Since $\mathcal{N}$ can be of the order of one million for flows around basic wings, this implies that even if we can solve the linear system efficiently, storage requirements will be a limiting factor. For the two factor method only the matrix for one spanwise slice or one line in the spanwise direction need be stored at any one time. This has the effect of reducing the matrix storage requirements at any one time in the calculation to $max(225\mathcal{N}_{slice},125\mathcal{N}_{line})$ where $\mathcal{N}_{slice}$ is the number of grid points in a spanwise slice and $\mathcal{N}_{line}$ is the number of grid points in the spanwise direction. Since $\mathcal{N}_{line}\mathcal{N}_{slice}=\mathcal{N}$ it can be seen that the storage requirements have been reduced substantially (by around two orders of magnitude for the test case examined in this paper).

As a test case we shall consider flow over the ONERA M6 wing in transonic conditions. The experimential data for this wing is available in [13] with several previous computational results including those in [11]. The flow problem we consider here has a free stream Mach number of 0.84, an incidence of 6.06 and a Reynold's number of 11 million. For this case 250 explicit steps were required before FUN was used with a CFL number of 10. The residual is reduced about 4 orders of magnitude from its initial value. This was also observed for flows over aerofoils in [8] and was due to small oscillations in the pressure at the far field.

The comparison of the computed pressure distribution with the experimential results of [13] at six spanwise slices are shown in figure 3. Good agreement is obtain for the flow except for the position of the shock and the very last station at 99% span. This has also been observed in [11] for this test case. Shock induced separation occurs after the strong shock near the tip and the Balwin Lomax model is known to be inadequate for this phenomenon. In [11] the Johnson–King model was also implemented which significantly improved the results. Figure 3 shows that mesh refinement in the streamwise direction has very little effect on the solution apart from sharping the strong shock near the tip. However refinement in the spanwise direction not only improves the resolution of the tip of the C-H grid and hence the pressure distribution before the shock close to the tip; but also the strength of the first shock in the mid span region. This can be more clearly seen from the upper wing surface pressure contours shown in figure 4.

4 Parallel Implementation

A detailed description of the parallel implementation of the 2 and 3-D methods can be found in [6]. In the present section we summarise the main features and give sample results.

The major obstacle to an efficient parallel implementation of the AF-CGS method is the inherently sequential nature of the ADI procedure. This was overcome in [1] by using a transposition of the data to allow complete ADI sweeps to proceed independently on each processor. We use this approach here although extra communication is required for the present method because of the matrix-vector products required in the CGS algorithm.

The computational space is mapped onto the nodes by grouping complete mesh lines in both the ξ and the η directions onto a single node. Care has to be taken to make sure that ξ lines on either side of the wake cut are mapped to the same processor. The computation then falls into three phases.

First, the matrix is generated and the factors are put in upper triangular form. The next phase is the multiplication of a vector by the matrix and finally we have multiplication of a vector by the preconditioner which reduces to back substitution on the triangular factors of the ADI factorisation. For each phase data is held on a node for complete lines in one direction in the mesh and the entire computation relating to that direction is completed. The data is then communicated so that information for complete lines in the other direction is held on a single node and the computation for that direction proceeds.

The parallel code was also implemented on a cluster of Silicon Graphics Indy workstations at the University of Glasgow. The message passing was accomplished by using PVM version 3.3. The comparison of algorithm speeds (time in μsec/grid point/time step) on the SGI cluster is shown in table 1. The results where obtained on a coarse mesh with only 2400 points and hence the loss in efficiency is quite small when this is considered.

Machine	algorithm speed	efficiency
SGI cluster 1 nodes	958	1.00
SGI cluster 6 nodes	230	0.69
SGI cluster 8 nodes	194	0.62

Table 1. *algorithm speeds in μsec/grid point/time step on the SGI cluster.*

The three-dimensional algorithm has two distinct phases. First, there is the generation and solution of the large linear system arising from each spanwise slice of the mesh. Secondly, there is the solution of the banded linear systems arising from the second factor in the spanwise direction.

The first phase is split between processors in two ways. First, the spanwise sections are split into groups. Each group is then assigned to a set of processors with each spanwise slice in the group being treated in a similar way to the two dimensional algorithm described above by those processors. The communication between the different groups of processors, each treating a different set of spanwise slices, is simply that which would be required by an explicit method so that the contributions to the residual (or the right-hand-side of the linear system) from the spanwise fluxes at the interfaces between the spanwise groupings can be evaluated. Since there is significantly less communication involved at this stage than is required to solve a spanwise slice in parallel, it is clear that the most efficient partition of the problem will arise when as large a number of spanwise groups as possible is used. For a fixed number of total processors this will reduce the number of processors which operate on a spanwise section.

The second phase of the calculation involves assigning complete spanwise lines in the mesh to single processors. Again, a transposition of the data is used so that the calculation involving a single line can proceed on a single processor without further communication. Once the updates are available a second transposition is used to restore storage by spanwise slices for the next time step.

The method has been implemented in parallel on a range of machines. The algorithm speeds for the Cray T3D and the SGI cluster are given in table 2 for grids with 140000 grid points for the T3D and roughly half this number on the SGI cluster. The parallel efficiencies will increase when the grid is refined, however a high parallel efficiency has been obtained on 128 nodes, even for this relatively small problem. Excellent efficiency is obtained on the SGI cluster.

No. of nodes	Explicit timesteps		Implicit timesteps	
	speed	efficiency	speed	efficiency
T3D 1	417	1.00	1510	1.00
T3D 16	29.6	0.88	107	0.88
T3D 32	15.6	0.84	55.0	0.86
T3D 64	8.25	0.79	28.9	0.82
T3D 128	4.63	0.70	15.7	0.75
SGI 1			2372	1.00
SGI 6			416	0.95

Table 2. Algorithm speeds in μsec/gp/ts and parallel efficiency for the Cray T3D and SGI cluster

5 Conclusions

The programmes that are providing a world class computing environment for the development of CFD codes at the University of Glasgow were described. A high quality access to the 320 processor EPCC Cray T3D was obtained through forming the CCAF consortium on the problem targeted in this report. At the other end of the cost scale, the development and description of a parallel environment based on the spare capacity on workstations mounted on a quality network under the HNW project was described. It was predicted that this latter type of computing environment would be a standard within the design offices of Aerospace Companies in the future.

An implicit method for simulating three-dimensional compressible and viscous flow developed to run on a distributed memory parallel environment is outlined. The AF-CGS method is based on a two-dimensional approach which consists of an iterative solution of the linear system by the conjugate gradient squared algorithm with preconditioning by the alternating direction implicit factorisation. The FUN (factored-unfactored) method tackles three dimensional flows and builds on the two dimensional method by factoring the linear system into a factor arising from spanwise slices in the mesh and a block penta-diagonal factor arising from strips in the spanwise direction. The more complicated factors arising from the spanwise slices are solved by the two dimensional method. This approach yields a method which has similar properties to the 2-d ADI method, a situation which is substantially better than a three dimensional version of ADI. A study concerning the optimisation of the AFCGS code using RAE 2822 Case 9 was carried out. Three levels of mesh sequencing were used to obtain a starting solution on a fine mesh of 257 x 65. Then the optimal CFL number used was increased to 100, and the overall time to converge to within 0.25 per cent of the fully converged lift value was reduced by a factor of 5. When applied to unsteady flows AFCGS was shown to allow for larger time steps and a reduced computational cost when compered to ADI.

The FUN code was tested through the prediction of the flows over the ONERA M6 Wing using the Cray T3D. Even for the relatively course grid tested parallel efficiencies of 75 per cent were achieved using 128 nodes. Improved efficiencies will be achievable using finer grids. The comparisons with the experiment using a Baldwin-Lomax turbulence model were found to be satisfactory, but improvements are expected when the k-ω turbulence model is implemented.

Future work includes the development of multi-block approach and the testing of the unsteady 3-d code and its coupling with a structural code to tackle aeroelasticity cases. Work is underway on multiblock extensions of the methodology presented.

ACKNOWLEDGEMENTS

This work has been carried out with the support of the Ministry of Defence, the Engineering and Physical Sciences Research Council, British Aerospace and the Joint Information Systems Committee of the Joint Higher Education Funding Councils under the following grants: EPSRC/MOD GR/H47371, DRA/MOD/BAeFRN1C/407, EPSRC GR/K42264, NTI/65. The authors would like to thank Mark Woodgate for obtaining the three dimensional results shown in this paper and to Dr Ian Glover for his contribution to the early part of the work. The work on the cluster has been carried out by Bill McMillan, Dr Xiaokun Zhou and Angus McCuish. The mesh generation subroutines were supplied by Dr. A. L. Gaitonde of Bristol University.

REFERENCES

[1] T. Chyczewski, F. Marconi, R. Pelz, and E. Churchitser, 'Solution of the Euler and Navier-Stokes equations on a parallel processor using a transposed/Thomas ADIalgorithm', in *11th AIAA Computational Fluid Dynamics Conference*. AIAA, (1993).

[2] B.E.Richards et al, 'Computation of complex aerodynamic flows - CCAF project', Technical report, Technical Annex to Proposal to EPSRC (unpublished).

[3] K.J.Badcock, 'Newton's method for laminar aerofoil flows', Aerospace Engineering Report 30, Glasgow University, Glasgow, UK, (1993).

[4] K.J.Badcock and A.L.Gaitonde, 'An unfactored method with moving meshes for solution of the Navier-Stokes equations for flows about aerofoils', *submitted for publication, August,1994*, (1994).

[5] K.J.Badcock and B.E.Richards, 'Implicit time stepping methods for the Navier-Stokes equations', in *12th AIAA CFD conference, San Diego*. AIAA, (1995).

[6] K.J.Badcock and B.E.Richards, 'Implicit time stepping methods for the Navier-Stokes equations', *to appear in AIAA Journal*, (1995).

[7] K.J.Badcock, I.C.Glover, and B.E.Richards, 'Convergence acceleration for viscous aerofoil flows using an unfactored method', in *Second European conference on CFD*, pp. 333–341. ECCOMAS, (1994).

[8] K.J.Badcock, I.C.Glover, and B.E.Richards, 'A preconditioner for steady two-dimensional turbulent flow simulation', *submitted for publication, May 1994*, (1994).

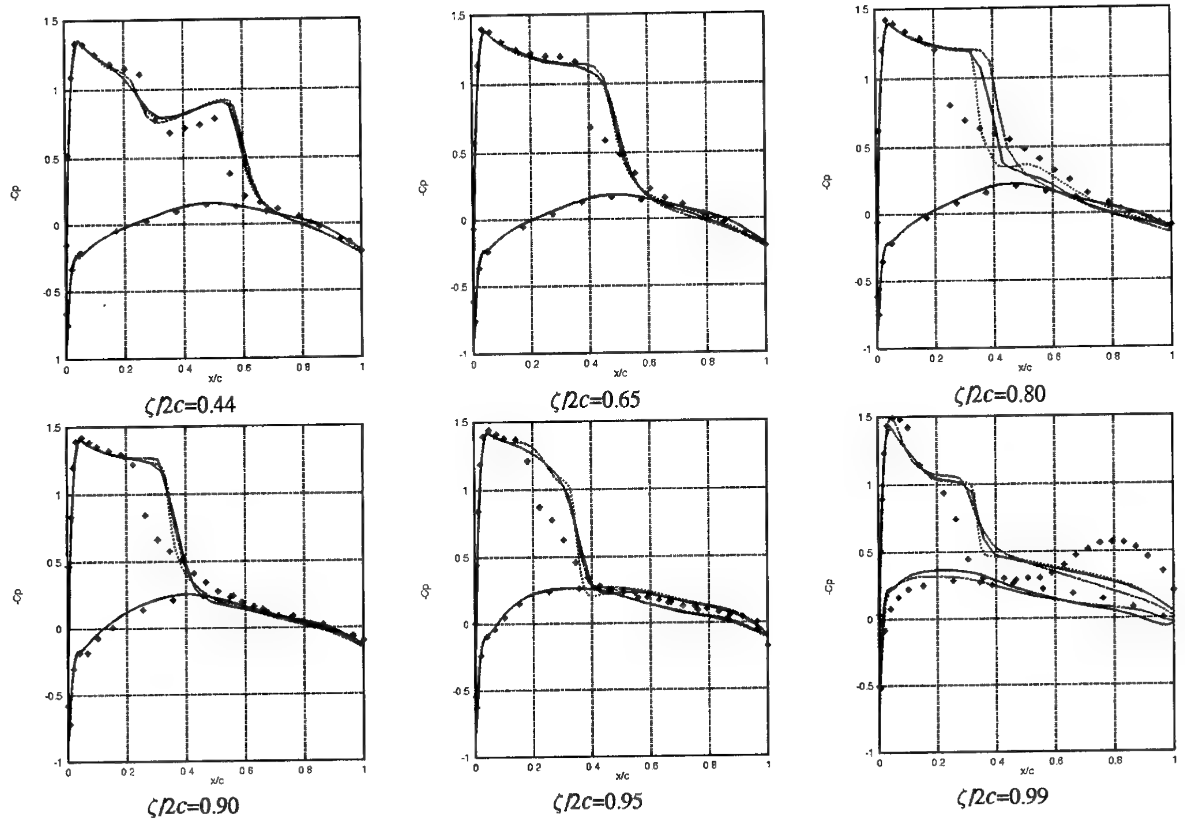
[9] L.Dubuc K.J.Badcock, X.Xu and B.E.Richards, 'Preconditioners for high speed flows in aerospace engineering', *to appear in Numerical Methods for Fluid Dynamics V*. Institute for Computational Fluid Dynamics, Oxford, (1995).

[10] P.Sonneveld, 'CGS: A fast Lanczos-type solver for nonsymmetric linear systems', *SIAM Journal Statistics and Computing*, **10**, 36–52, (1989).

[11] R. Radespiel, C.Rossow, and R.C. Swanson, 'Efficient cell-vertex multigrid scheme for the three-dimensional Navier-Stokes equations', *AIAA Journal*, **28**, 1464–1472, (1990).

[12] M. Vitaletti, 'Solver for unfactored schemes', *AIAA Journal*, **29**, 1003–1005, (1991).

[13] V.Schmitt and F.Charpin, 'Pressure distributions on the ONERA-M6-Wing at transonic Mach numbers', Technical Report AR-138, AGARD, (1979).



$\zeta/2c$=0.44

$\zeta/2c$=0.65

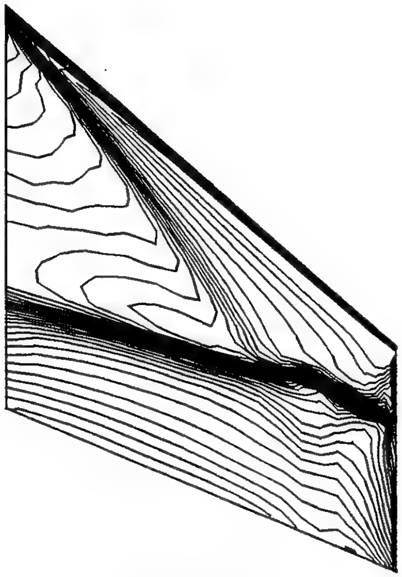
$\zeta/2c$=0.80

$\zeta/2c$=0.90

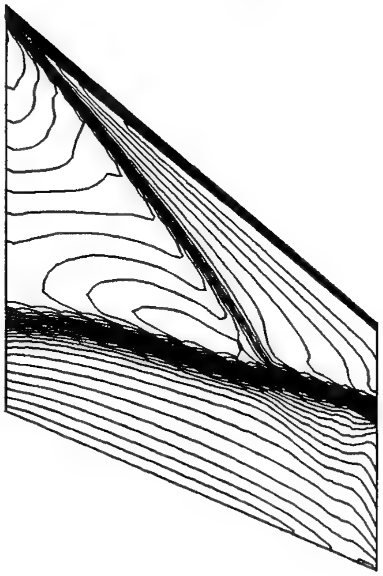
$\zeta/2c$=0.95

$\zeta/2c$=0.99

Figure 3. Comparison of computed pressure distribution with experiment for ONERA M6 wing :- Solid line 129 × 33 × 33, Dashed line 129 × 33 × 97, Dotted line 257 × 33 × 33



129 × 33 × 33 grid.



129 × 33 × 97 grid.

Figure 4. Surface pressures for 6.06 degree ONERA M6 wing.

PROGRESS AND CHALLENGES IN CFD METHODS AND ALGORITHMS
GENERAL DISCUSSION

J.W. Slooff, NLR, Netherlands
After Dr. Kroll has given his opening remarks we will open up to the floor and try to get a, hopefully, lively discussion on various issues and aspects that we may wish to address. But first, Dr. Kroll, please give us your on-the-spot evaluation.

N. Kroll, DLR, Germany
Thank you for the invitation. I appreciate that I can act as the evaluator for the CFD Symposium. Before I go into details, I would like to mention that this evaluation reflects my personal thoughts and is based mainly on the oral presentations. Only a few papers reached me before the Conference, so I did not get the time to go into the details of the papers. Therefore, in the written version some of my statements may be revised, but I think the essential messages will not change.

The background of this Symposium is the fact that CFD, as we all know, is widely accepted as a key tool for aerodynamic design. However, on the other hand, we also know that CFD still has deficiencies in accuracy, complexity, robustness, and efficiency. Due to this, in industry CFD is not yet being exploited as effectively as one would expect. Therefore, this Symposium has been set up with the aim to present and discuss those topics which are considered as likely to constitute pacing items and new challenges in CFD. The work presented here will be evaluated against the ambitious theme of the Conference. From my point of view and from what I saw in the invited papers, for the aeronautical industry CFD is expected to deliver: 1. detailed viscous flow analysis for complex geometries at high Reynolds numbers, 2. accurate prediction of aerodynnamic data, 3. fast turnaround calculations at acceptable costs, 4. aerodynamic design and optimization of aircraft components or complete aircraft and 5. interdisciplinary analysis. There may be many other key problems, but in my opinion, these are among the most important ones in order to raise the confidence level of CFD in the aeronautical industry.

With respect to the scope of the Conference, I expected contributions to the following topics: improvement of basic algorithms including space discretization, time integration and fast iterative methods; advanced techniques to treat complex configurations including blockstructured methods, unstructured and hybrid grids as well as Cartesian and Chimera techniques; adaptive methods; parallel computing; effective algorithms for more complex applications such as turbulent flows, chemically reacting flows and unsteady flows; design and optimization methods; effective methods for miltidiscipline physics.

Three invited and 34 technical papers were presented. First I would like to make some general remarks on the technical quality of the papers. Many of the papers were of high quality because they represented the current status of the CFD community and they identified or presented new important directions of algorithmic development in CFD. But in my opinion, also many papers of lower quality were delivered which either were not within the scope of this Symposium or did not represent the current status and progress of CFD. Moreover, some of them reinvented well-established knowledge in CFD. The Symposium covered eight major topics as shown in this vugraph.

Although many papers addressed several subjects, I categorized them according to their central focus. The result of this classification is somewhat different from the session grouping which was set up by the Program Committee. There were 10 or 11 papers on advanced discretization schemes, only three papers on fast implicit iterative solvers and a bunch of papers on parallelization. Several papers were given on unstructured meshes, overlapping grids and hybrid grids. This morning we heard papers on adaptive grids and two papers concerning specific algorithmic aspects on chemically reacting flows. We had some papers on DNS/LES and on unsteady flows. In the following, I would like to go through each subject and to make some comments on what was presented and whether the major challenges were addressed by the papers.

First let me say a few words about the invited papers. The first paper was presented by Anthony Jameson. He gave an excellent overview about the present status, challenges and future development of CFD. He identified some important challenges, in particular the 3-D viscous flow simulation for high Reynolds numbers. He mentioned that about 8 to 10 million points are needed in order to accurately resolve turbulent flows and to predict the drag coefficient within one count. His presentation on the unified theory for 1-D shock capturing methods was very interesting. He showed that unifying different schemes may help in designing new improved methods such as his newly developed CUSP or HCUSP scheme. Jameson also addressed the important topic of aerodynamic design and optimization.

The second invited paper was delivered by Paul Rubbert. He talked about CFD research in the changing U.S. aeronautical industry. I think it was a very interesting paper because he identified the challenges which are beyond the technical ones. He made an analysis of the process by which CFD capabilities are created. He mentioned that research can be improved by introducing new principles like customer focus and customer satisfaction. From the technical reviewer's point of view, some comments and statements of the aeronautical industry on the status of CFD and future requirements would have been desirable.

The third paper was presented by Doyle Knight. He gave a nice overview on parallel computing. Since he explained the terminology used in parallel computing, he formed the basis for the audience to follow the technical papers on parallelization. He discussed important issues, and he gave several examples of experience with parallel computing in the aerospace industry.

From my point of view, the Program Committee did a good job concerning the selection of the keynote papers. In my opinion, however, an invited paper on status and progress on grid generation for complex configurations was missing. Although grid generation was not a subject of this Symposium, I think that such a paper would have been very helpful for the assessment of structured and unstructured methods. The issues of turnaround time and accuracy of a numerical method very often depend on the capability of the available grid generation procedure.

Now let me come to the specific subjects. Several papers on parallel computing were presented. It is obvious that routine use of CFD and future large applications in aeronautics require parallel computing. The papers addressed several important issues such as parallelization strategies, portability, performance and load balancing. Some papers were devoted to the adjustment of algorithms designed for sequential computer to parallel architectures. The issue of scalability was only barely addressed although it is one of the key features for efficiently exploiting parallel computing. Only a few 3-D applications on parallel computers have been presented. The efficient use of parallel architectures for 3-D complex industrial configurations seems to be still a major challenge. The reason for this may be the problem of load balancing. In the case of unstructured meshes, much work has been done with respect to domain partioning and some public domain software is already available. However, for structured meshes the load balanced partioning is much more complicated mainly due to the geometrical restrictions. Furthermore, I believe that the adjustment of sequential algorithms to parallel computers is not sufficient. New parallel algorithms have to be developed. As an example, it is well known that the multigrid method is not fully scalable, and therefore may not exploit the full performance of massively parallel computers. In conclusion, challenges for parallel computing for the near future are scalable implementations of 3-D applications, load balancing for structured mesh calculations and development of new parallel algorithms. Future work should address these issues.

Now let me discuss the topic of advanced space discretization. Various promising schemes have been presented showing some improvements over conventional methods. For example, papers were devoted to quadratic reconstruction with flux-limiters, improved flux-splitting schemes, multidimensional upwinding and kinetic flux-splitting. My criticism here is that in some of these papers, the assessment of the new algorithms were restricted to only one or two aspects of spatial discretization. Designing new discretization schemes, several different aspects have to be addressed, including high resolution of viscous shear layers, sharp shock resolution, conservation, robustness at shocks and in expansion regions, overall efficiency and compactness of the stencil. In my opinion, a detailed comparison of available schemes covering all these issues is needed in order to assess potentials and limitations of advanced algorithms. I also would like to mention that very often a detailed accuracy assessment of new or modified methods is not carried out. An assessment study should include investigations with respect to grid refinement and other important numerical sensitivities. Well-established test cases for the Euler and Navier-Stokes equations should be calculated to raise the confidence level of the new techniques. Of course, the advanced methods have to be applied to those problems for which the standard schemes show substantial deficiencies. Multidimensional upwinding, from my point of view, made large progress in the last few years, but I still think that these schemes - and also kinetic algorithms - are not yet at the stage to be used in a 3-D production code. The major challenge for advanced discretization schemes is the accurate calculation of 3-D viscous flows. Beside high resolution schemes, improved turbulence models are required. Turbulence modelling, however, was not a topic of this Symposium.

Let me come to the third subject on fast implicit and iterative methods. Here good papers on Newton-Krylov subspace methods were presented. For standard cases, like 2-D inviscid flows or viscous flows with moderate Reynolds numbers, the more sophisticated methods such as multigrid, Newton-Krylov subspace methods and advanced implicit schemes perform almost equally well. From the literature, it is obvious that multigrid is mostly used with structured meshes, whereas for unstructured grids, very often Newton's method with Krylov subspace iteration is applied. For 3-D flows around complex configurations the situation is not clear. We do know that for generic configurations and moderate Reynolds numbers multigrid is quite efficient. There is not much known about the Newton-Krylov methods. Open questions are the subspace dimension, the memory requirements and the computational costs. Much more effort is required to explore the capabilities and limitations of these methods. In my view, the real challenge concerning the development of efficient time integration algorithms was not addressed here. It is the simulation of realistic Reynolds number flows in 2-D and 3-D. Due to efficiency reasons, for these flows high aspect ratio cells are required. Due to the lack of efficient smoothers, the convergence behavior of the multigrid method gets worse. In the case of Krylov subspace methods, a suitable preconditioner has to be designed. Future work should be devoted to the development of efficient time integration algorithms for stiff discrete equations due to high aspect ratio cells.

There was a nice paper on time-preconditioning, which I think is a very interesting approach to achieve Mach number independent convergence. A key concern is the development of a unified flow solver covering incompressible flows up to hypersonic flows. I do not think that the technique is already mature, however, preconditioning is a good candidate to reach that goal.

Another subject dealt with unstructured and hybrid methods. The paper from Rockwell stated that unstructured grids are well suited for inviscid flows including flows around complex 3-D configurations. On the other hand, experience shows that for accurate viscous calculation, some kind of regular cells are required in the boundary layer. So the approach of hybrid grids may be a good choice because it combines all the advantages of structured and unstructured meshes and offers the possibility for an automatic simulation of 3-D complex configurations. The work presented here on hybrid meshes is in an early stage and substantial effort is required to establish a valuable tool for complex viscous applications. For the simulation of configurations with moving bodies, the overlapping grid technique seems to be very interesting. The meshless technique approach is an interesting idea, but it shows many deficiencies. For example, conservation is not guaranteed and the control of accuracy is quite difficult. Much work is required to get some confidence in this approach. For all discretization strategies, considerable effort is still required to significantly reduce the turnaround time for viscous simulations for complex 3-D configurations. Some promising results have been presented here.

The next subject covers the activities on adaptive methods. It is well known that adaption is an important issue for cost-effective calculations. Various strategies have been presented including mesh movement and mesh refinement for both structured and unstructured grids. In my opinion, there are several open questions, some of them were addressed at the Conference. A key issue is the selection of suitable criteria for grid adaption. As proposed by several papers, finite element error indicators seem to be the right choice. They ensure that the solution will not be sensitive to the adaption pattern. However, so far in most applications local flow gradients are used as sensors. In these cases, the estimation of the overall accuracy is quite difficult and a grid independent solution may not be obtained. With respect to parallel computing, the problem of dynamical load balancing occurs, especially in the case of structured meshes. The challenge of adaptive methods is the application to 3-D viscous flow fields. As mentioned here by several authors, considerable work is required to extend error based indicators to viscous flows.

Concerning unsteady flows, several papers presented time accurate calculations for incompressible and compressible flows. Some attempts were made to cut the cost of time accurate calculations. However, it is obvious that new innovative concepts have to be developed in order to efficiently simulate 3-D viscous unsteady flows.

A few papers addressed LES and DNS. At the moment both simulation techniques focus on fundmental research of flow physics, especially turbulence. Specific requirements on LES and DNS solvers were discussed including high resolution in time and space, adaptive grids and parallel computing. Based on these sophisticated methods, one paper held out a prospect of large eddy simulation of a clean wing at moderate Reynolds number in the near future. In my opinion, significant research work on both algorithms and subgrid model is required to enable this simulation. Some of the papers dealing with this subject did not meet the scope of this Symposium.

The topic of chemically reacting flows was covered only by two papers. Both presented modifications and improvements of numerical methods to meet the requirements of hypersonic reacting flows, namely, sharp capturing of strong shocks, high resolution of viscous regions, robustness in regions of flow expansion and efficient solution of stiff equations. Promising results for 2-D and 3-D flows were presented.

This concludes my technical comments on the various subjects. Now I would like to give a few concluding remarks. Measured against the theme of the Symposium, in my opinion, many papers of high quality but also many papers of lower quality were given. Concerning the technical standard, there was quite some difference between this Symposium and other conferences such as the AIAA conferences in the U.S. or the ECOMASS in Europe. In order to improve the quality of the papers, one should ask for extended abstracts. CFD has so many aspects and facets that it is very difficult to assess the quality of a paper with only a few pages of abstract. Furthermore, one should define some criteria or certain procedures which have to be met by the abstracts. This could include the calculation of specific test cases.

Nevertheless, I would like to say that all in all the Symposium was interesting. We saw some recent developments and achievements in CFD which I have mentioned before. Several problems were identified, being pacing items for algorithmic improvements and new developments. However, from my personal point of view, the Symposium did not reflect the actual status of the CFD community compared to other CFD conferences. Many leading experts were not present. Especially, there was only a small contribution by the U.S. Many aspects and recent developments were not addressed in this Conference. Furthermore, in some areas, I think CFD is much more developed than it was presented here. I would like to say a few words concerning the challenges I have mentioned previously. Many of the key issues important for industry were not covered here. No paper tackled the problem of high Reynolds number flows. There were not many papers on accurate drag calculation for viscous turbulent flows. Concerning the problem of short turnaround time for complex configurations, some advanced approaches including unstructured and hybrid methods were presented. However, no paper addressed 3-D viscous calculations around more complex geometries. There was only one paper - Jameson's invited lecture - dealing with design optimization. No paper addressed interdisciplinary methods, an issue which is definietly a future challenge in CFD.

Finally, I would like to remark that, in my opinion, the scope of the Symposium was too encompassing. It is almost impossible to cover all important new directions in CFD within an AGARD conference of three and one-half days. It would have been better to restrict the Symposium to some specific subjects, say adaptive methods. In that case a comprehensive overview and review of this particular subject would have been possible. New directions and developments and their critical assessment could have been addressed in more detail.

J.W. Slooff, NLR, Netherlands
Thank you very much Dr. Kroll for what I think was a very appropriate and to-the-point evaluation with a good balance of critique and praise. One small remark from my side, I think that there is an internal conflict between two of your statements in the sense that on the one hand you think not enough subjects of CFD were covered and on the other hand, you said the scope was too wide for only three and one half days.

N. Kroll, DLR, Germany
Measured against the scope and aim of this Symposium, not all important issues and subjects were covered.

J.W. Slooff, NLR, Netherlands
We are now at the Open Discussion part of the final session. The last thing I would like to do is to put the discussion into some sort of a straight jacket, but on the other hand, it occurred to me that if we go about this in a completely unstructured way, the discussion might quite easily become pointless. What I suggest is that we proceed in the following way. First of all, I would like to spend 5 or 10 minutes to give direct comments on some of the statements made by the Technical Evaluator. Some of you may have the urge to do so. After that, I suggest that we try to look at things from some distance in order to get a better perspective of what we are doing and what we are doing it for. In doing so, I think, we should add a background as a sort of framework against which we can project our comments, remarks and suggestions. On this background we should keep questions in mind like: what are industry's requirements? what do we have to offer as the CFD research community, what kind of new developments?, which of these developments have the best prospects for better meeting industry's requirements.

Before we start the actual discussion, I have to point out a few administrative things. This Round Table Discussion is being recorded and a transcript of the tape will be made. That does not mean that you should confine yourself. You don't have to be afraid of saying things that you will be confronted with afterwards. You will be sent a copy of the transcript of the tape and you will have the opportunity to edit your comments and remarks. In order to be able to do so, it is necessary that you clearly state your name and affiliation so that we know who spoke and who to send the transcript to. I will come back later on some of the basic questions and provide a little bit more framework for our discussion. But first, who would like to give some direct comments on some of the remarks that the Technical Evaluator gave us just a few minutes ago? I imagine the Program Committee Chairman might have to say something.

J.A. Essers, University of Liege, Belgium
First of all I think that Dr. Kroll did a very good job, and perhaps you will be surprised to note that I agree with almost everything he said. But anyway, I have a technical remark to make on one point. This point is concerning the fact that it is better to use a structured grids on viscous flows, and perhaps an unstructured grid is good for inviscid flows. I disagree with that. Many people believe that with unstructured grids, you cannot compute accurately a boundary layer or other shear layers. That is wrong. We made some calculations with quadrilateral and triangular grids that were extremely irregular and got very good accuracy, with for example, parabolic re-construction techniques. My feeling is that many people use discretizations which are not accurate enough for the viscous terms when the grid is distorted, but if you use schemes which are weakly sensitive to mesh distortions, it can work fine. Anyway, I must confess that you usually need more points in a boundary layer if you use unstructured grids then if you use a structured grid, so it

is anyway worthwhile to use the hybrid grids for that reason. Now I would like to make another comment concerning Dr. Kroll's remarks. I said that I essentially agree with what he said, but he should be aware of the fact that the Program Committee of an AGARD meeting has constraints that organizers of large conferences don't have. For example, we are almost not allowed to have parallel sessions. And parallel sessions would have been necessary. I propossed parallel sessions, but I had immediately 10 opponents in the group, so it was impossible to make it. That is namely, because we need a Technical Evaluator who cannot attend all sessions at the same time of course. Something else, also, is that we have some, let us say, political constraints, in the sense that it is important to AGARD that all NATO countries can participate in such conferences, to present the status of the research in their country, and that is a constraint other conferences don't have.

B. Masure, STREHNA, France
The Technical Evaluator said that many papers did not address the problem of the accuracy assessment. I ask the Technical Evaluator to say to us what is exactly an accuracy assessment for a code.

N. Kroll, DLR, Germany
For structured meshes, one should make sure that by refining the grid the results will become independent from the grid. Furthermore, the order of accuracy claimed by theory can be checked by grid refinement studies. In case of unstructured meshes the accuracy assessment may be more difficult, but I think we can borrow some techniques from finite element theory.

P.W. Sacher, DASA, Germany
Six years ago we had a big Symposium on the subject of Code Validation, CFD Validation. This was a subject that I missed here, specifically in your remarks, in your evaluation. Does it mean that code validation is no longer an issue for CFD?

J.W. Slooff, NLR, Netherlands
I am pretty sure that Dr. Kroll is going to say that it is still an issue, but that it was, on purpose, outside of the scope of this Conference. Are there any further direct comments on the Technical Evaluator's remarks?

F. Mokhtarian, Canadair, Canada
I just wanted to say that I agree with most of the comments of the Technical Evaluator regarding the quality of the Conference and the papers. However, the comment I would like to make is that I thought his comparison of the Conference with some of the other conferences was perhaps a bit unfair. This isn't the first time I have attended an AGARD Conference. I have been to many other conferences in Canada and the U.S. and you always get a variety of papers, you can't always be very critical. It is very difficult to tell the quality of some of the papers ahead of time. There were some papers I thought perhaps were not exactly up to par, but there were lots and lots of papers that were very high quality and I was glad I was able to attend. The only comment I wanted to make is that I think he was a

bit harsh comparing this Conference with some of the other Conferences.

N. Kroll, DLR, Germany

Essentially, I think you are right, but you have to read again the title of the Conference. The Conference has the title, "Progress and Challenges of CFD Methods". This is an ambitious title. You have to make sure that most of the papers of the Conference will meet the high demands of the Symposium.

J. van Ingen, Delft University, Netherlands

In regard to the remark by Prof. Essers about the boundary conditions on an AGARD meeting I think we as a Panel have to think about what especially is the place of AGARD in all of these CFD conferences. I think CFD conferences organized by AGARD should have some specific ideas in it to bring together the users and developers and maybe we should leave the more fundamental subjects to a special conference on that. The present criticism may be due to the boundary conditions we have to put on AGARD conferences.

J.A. Essers, University of Liege, Belgium

Just to see if we agree, let me just make a few comments and ask a question to Dr. Kroll. I think that, as you said, there were very good papers in this Conference, there were some that were not as good, of course. Unfortunately, very few papers here had a sufficient vision of the future. But that remark is also valid for many conferences. Nevertheless, I agree that the title of the conference was perhaps too ambitious, but we couldn't know that before receiving the submitted abstracts. I also would have liked to hear more people saying why they use this technique instead of another; why it is more appropriate because in the future they want to tackle that problem and that problem. For example, I would have liked somebody making an Euler calculation, explain that he uses that scheme instead of another one because he thinks that that scheme will be better for future viscous flow calculations. Nobody discussed such issues. Is it perhaps what you want to say Dr. Kroll?

N. Kroll, DLR, Germany

This is exactly what I wanted to say. I tried to define challenges on the different subjects which should be addressed by the CFD community.

A.G. Panaras, HAF Academy, Greece

I think that it should be more appropriate to state that progress has been reported on some new ideas and not to make the distinction between good and not good papers. Many authors have made substantial efforts in preparing their work and certainly there is always something new that comes in a conference like this.

J.W. Slooff, NLR, Netherlands

Now with your permission I would like to switch to the second part of this discussion. I would like to get you thinking, if not talking, about three key questions that we have to deal with and that we have to get answers for. To trigger the discussion a bit further and to perhaps provoke you a little bit into making comments, let me try to list what I think are industry's three most important requirements. One is to increase the confidence level of the codes, which means, from my point of view, that for every application you would like to know what the accuracy is. I don't think there are many codes that, together with the CP distributions and what have you, provide an estimate of the accuracy. I think that is something that we should strive for. Robustness, that is clear, is another aspect. Reduction of the problem turnaround time has also been mentioned by Dr. Kroll, of course. Here grid generation is the bottleneck, in particular for the structured grid approach. That leads us to efficiency. We may loosely define efficiency as accuracy divided by cost, and cost is more or less proportional to time. We can distinguish between manpower time needed, particularly for preprocessing, that is geometry handling and grid generation and the pure computer time.

I don't think the post processing part of it is a big deal here. If we look at that formula and address the different parts in it, we know that accuracy is in the first place a function of the physical model, including the turbulence model (that was on purpose not addressed here at this Conference). The other important parameters are the number of grid points, the distribution of the grid points, the "order" of the method plus the artificial dissipation models and whatever flux upwinding or multidimensional upwinding scheme is used. On the cost-side we have prepocessing as I already mentioned, plus the CPU time. For the latter, the number of grid points is again a parameter, plus the "numerical" scheme, the solution argorithm, and of course, the hardware. The latter is, however, beyond the scope of this Conference.

What I would like to do is to discuss the current main developments in CFD against the background of industry requirements, the efficiency requirement in particular. If, as a baseline, we take the currently well-established multi-block structured type of codes with conventional types of schemes, let us say the Jameson Flo-57, -67 level of technology, we can try to estimate where we can improve, relative to that baseline situation. For the unstructured grid approach we may, for the same number of grid cells, have perhaps somewhat less accuracy than for a structured grid solution. I am not completely sure about that, and you may wish to comment. However, there is certainly a lot of gain in the grid generation part. If we look at adaptive grids, it is clear that unstructured as well as structured adaptive grids have a potential for increasing the accuracy for a given number of grids cells. However, grid adaptivity also has the prospect of reducing the preprocessing and the grid generation calendar time. This because with adaptation the first grid you start with doesn't have to be as good as is the case when you do not have an adaptive grid approach. The CPU aspect for given accuracy is also clear. I think the biggest advantages for adaptive grids are in the unstructured case. I think there we have a bigger potential for gain in accuracy for a given number of grid cells or reduction of the number of grid cells for given accuracy than in the case of structured grids. Higher order schemes, multi-dimensional upwinding and similar refinements, are, of course, good for accuracy. I am not quite sure what it means for grid generation. I have the feeling that some of these more subtle schemes may require better meshes than more conventional schemes. They also may require some more computer time, at least for the same number of grid points. However, because the number of grid points required for a certain accuracy level may be less, we may still gain something. I don't know what the balance is. You may wish to comment on that.

One further remark, on adaptive grids. We might wonder what is more efficient - to implement a highly sophisticated higher order scheme with the best thinkable multi-dimensional upwinding with only one grid point in the shock wave, or to have an adaptive grid scheme with, for example, two or three grid points in the shock. I am not sure which of the two is more efficient. I will stop here. This is just a little bit of provocation in order to get you out of your seats, so to speak. Who would like to shoot at this or anything else?

P.E. Rubbert, Boeing Commercial Airplane Group, U.S.
It is important to speak to how good do you have to be, what is the target. Not just faster, but how fast, etc. One of the things I seem to detect at this Conference is that many of the speakers had in their mind a different definition of the decimal point than I do. I heard talk about working hard on grid generation to reduce the time from three weeks to maybe one week or maybe one day. My experience in using CFD in an airplane design environment is that when you are talking about designing a wing, it wasn't too many years ago that that involved a sequence of about 75 full blown CFD runs, part analysis, part inverse design, etc. One day turnaround was unacceptable. We do not want to take 75 days to design wings. The decimal point belongs in terms of hours, not days. In our old design environment, our target was to get three turnarounds in an 8 hour day in the design environment. The challenge is now to reduce cycle time even more. So I think it is worth saying that some of the targets that I hear people setting for themselves will produce a capability which is not really acceptable and useable in a real airplane company environment.

J.W. Slooff, NLR, Netherlands
Thank you for that comment, and it reminded me that I forgot to mention one aspect in relation to high order schemes and accuracy. We are not looking for infinite improvements in accuracy. What we need is, for a given accuracy that we want to obtain, but not necessarily want to exceed, the highest efficiency, the shortest preprocessing turnaround time and the lowest CPU cost. Higher order methods usually have their greatest benefit if you require very high accuracy. If you have lower accuracy requirements they may not be so well suited for the purpose. In industry, you probably will agree with me, different levels of accuracy are needed in different phases of the design process. Industry is not always looking for the highest accuracy. That is something we also have to bear in mind in considering higher order methods.

P. Rubbert, Boeing Commercial Airplane Group, U.S.
The subject of accuracy - another thing I did not hear at the Conference was any discussion of CFD with respect to the environment in which we use it. I think it is very important that we learn how

to think about what we want from CFD in the presence of wind tunnel analysis and the other tools that we have for doing airplane design. For example, the question of accuracy. I heard many times people setting goals like we would like CFD to be able to calculate drag at this level of accuracy, and so forth, but the way we really do it in industry is that we don't depend on any one tool to give us the total answer. The total answer is arrived at by utilizing all of the information at your disposal; the information that CFD provides, the information that wind tunnels provide, your previous experience, etc. Integrate that all together into a judgement as to what something like the drag would be. Again, when we talk about accuracy of CFD, if it is going to take us 6 months to build a wind tunnel model and test it, that means one thing in terms of the amount of accuracy you need out of CFD. But I heard some discussion this week about stereolithography methods, and things like that that could lead you in the direction of what one might call overnight model manufacturing. If that happens in the wind tunnel, that has a major influence on the type of accuracy levels you would need out of CFD. If you could rapidly get a number out of the wind tunnel, maybe you don't need to focus so hard on CFD accuracy. I guess my point is we have to stop looking at CFD by itself. We have to learn to look at it with respect to the total environment.

D. Knight, Rutgers University, U.S.
I would like also to focus on this question of accuracy. As I have often understood it, it seems to be more of a question of accuracy as a function of resource rather than resource as a function of accuracy. Typically, for example, if you want to compute the total pressure recovery in an inlet in an industrial environment, the question is how long it will take to get within a certain accuracy. That may be 1% for the total pressure recovery, if it is a design, it may in fact be even smaller or perhaps larger. I think we yet, in the CFD community, don't focus enough on the question: given a level of accuracy of a particular type, like total pressure recovery, what is the resource required to get that. If you are in industry and you have a week to do a computation, can you actually predict the total pressure within 1%, or should you not try at all. Maybe that will take 2 weeks and that is the information that you need to know. This also raises the question of optimal design: the optimal design of your algorithm in terms of reconstruction of high order methods, and also the optimal design of your grid structure within that algorithm. That, of course, brings to the fore the question of an estimate for the accuracy of your scheme. That is an issue that was mentioned in a number of papers including the earlier one this morning by Friedrichs. In the CFD community we still do not yet have a good measure of accuracy, and how to predict that from our solution.

S.V. Ramakrishnan, Rockwell Science Center, U.S.
I have one comment on hybrid grids. From our experience in generating such grids, I can say that most of the difficulty lies in the region near the body surface for complex configurations. If you can develop a structured grid near the body, you might as well develop such a grid everywhere, because it doesn't take too much work to develop the grid away from the body surface. Therefore, if we

cannot solve the viscous problem with unstructured grid, we may as well not use it all. There is no point in using hybrid grids.

P.G.C. Herring, British Aerospace Ltd, U.K.
A comment first on adaptive grids, I am not sure if the size of the pluses and minuses is an indication of the potential benefit, but some of the work that we have been doing is beginning to indicate that in some of cases it is maybe not worth using adaptive grids. The time it takes you to develop the procedure and run the codes is often longer than it would take just running 2 or 3 cases of an ordinary grid. The other thing that surprises me is your last line on parallel algorithms which indicates that, for CFD fluid problems, there is not much benefit to be gained by going to parallelization. You have a small positive in the last column. As I say, if the size of the plus is an indication of the benefit, it appears we are not yet ready for parallelization with CFD.

J.W. Slooff, NLR, Netherlands
I hasten to say that I have not been very consistent with the sizing of the pluses. But I do think personally that a good grid adaptivity scheme is one of the most important things that we have to go after.

J.A. Essers, University of Liege, Belgium
I have two very specific questions. The first one is about chemically reacting flows. It is a question for Dr. Radespiel or Dr. Marmignon. Perhaps someone can answer it. Well, in the abstracts we received no proposals on the following subject. In the past, we expected that there would be some developments on techniques using different grids for different equations, for example, relatively coarse grids for the flow equations and the finer grids for the chemical reaction equations. I heard nothing on that issue in this Symposium. Is that idea still around or is it forgotten now?

The second question is concerning the DNS method. In the past, I expected that DNS would provide some kind of numerical wind tunnel or experiment to construct better turbulence models, classical turbulence modelling, or for example, for LES. Nobody addressesd that subject. My question is, at this time, are there some people who use DNS to try to construct better models for turbulence or not? Usually, when I attend a talk on DNS I hear nothing about that.

C. Marmignon, ONERA, France
I would like you to be more specific with the question if possible.

J.A. Essers, University of Liege, Belgium
In the past some people suggested, I namely think of Marsha Burger of the Courant Institute, but I am not sure, that some people in the U.S. were working in the field mentioned in my first question, i.e., the use of different grids when you have chemical reactions. Some of them have very short relaxation times, leading to very sharp gradients in a shock wave for example, so it could be worthwhile to discretize the kinetic equation corresponding to that reaction with a very fine grid and perhaps to use a coarser grid for another chemical reaction and still a coarser grid for the flow equations. So you could imagine to have a series of grids, three grids, for example.

Of course, the points of the coarse grid would also be points of the finest grid, like in multigrid techniques I would say, and for some equations you would only discretize them on the coarse grids and interpolate the results for the fine grids in order to save time. Is there some research going on in that field? Maybe it is not interesting, I don't know.

C. Marmignon, ONERA, France
We have not looked at this point.

J.W. Slooff, NLR, Netherlands
On the last question, that is the DNS, LES question, I think I saw three hands up there.

B. Geurts, University of Twente, Netherlands
The question you raised is, as exclusively mentioned, not within the scope of this Symposium. If you are interested in it, I would like to refer you to some of the work at Twente where we try to use DNS as a data base for developing subgrid models for LES which is an intermediate step for possible extension to Reynolds averaged turbulence modelling improvement. We are not unique in the world, there are several groups that have similar approaches in which they start from DNS.

P. Comte, LEGI, Institut de Mecanique de Grenoble, France
I think all the LES community has tested models in comparison with DNS, however DNS are currently restricted to fairly low Reynolds number flows. If we want to use LES for higher Reynolds numbers, maybe those comparisons wouldn't be that relevant.

N. Kroll, DLR, Germany
I just want to make a short comment on chemically reacting flows. In my opinion the most severe problem is the stiffness of the discrete system. I think you cannot overcome this problem by using different mesh types. You have to develop efficient algorithms to overcome that stiffness problem.

J. Jimenez, Escuela Superior de Ingenieros Aeronauticos, Spain
The question of the relationship between DNS and modelling is something that has been considered for several years. It is a question of what to expect. You cannot expect DNS to give you a model. That has to be done by modellers. What DNS gives you is "ground truth'. It gives you what the real flow is doing, and it gives you constraints on which models work and which ones do not. This has been practiced extensively now, at the CTR in Stanford, at Twente, as reported in this meeting, and at many other places. There are cases in which DNS is almost the only data available, as in the case of stress balances in separated flows, which are difficult to measure and difficult to model, but which have been computed with DNS. You can use those data to check whether a particular model works or not and, if it does not, it is up to the modeller to come up with a better one. That last step was, of course, outside the scope of the present meeting. DNS can do this, and it can give you some ideas of how to improve your model, but it cannot produce a model by itself. It is as difficult to get good models out of DNS as it has

been to get them from experimental data. There is nothing magic about DNS. It is just a better experiment.

J.A. Essers, University of Liege, Belgium
I don't know, but I suppose it is easier to get a lot of data from a calculation like DNS than from experiments. I don't know a lot in that field, but I would see DNS as a kind of experimental facility that can provide you with a lot of information if you need it.

J. van Ingen, Delft University, Netherlands
I think we should not forget that there has been a time, I refer to the Stanford trials in '68 and '80, when the idea was that we just had to wait for the ultimate turbulence model and all our problems would have been solved. Then, I think around the '80's, people started to realize that there is not a single turbulence model. You will have models for different kinds of flows. So if you say DNS is providing a different approach to experiments, yes, but you will need experiments, hence also these numerical experiments, in these different kinds of flow. Having a problem with calculating high Reynolds numbers will remain as long as you cannot do DNS for these high Reynolds numbers.

J.W. Slooff, NLR, Netherlands
I thank you all for your contribution to this discussion, and in particular Dr. Kroll for giving us the starting point for the Round Table Discussion. I think Prof. Essers would now like to formally close this Conference.

J.A. Essers, University of Liege, Belgium
I will make it very short. First of all, I would like to say that in my opinion, the Conference was satisfactory from several viewpoints. First of all in terms of attendance, if the attendance is a measure of the usefulness to the NATO people. I just would like to let you know that we had 124 attendees, including observers, Panel Members and authors. For those of you who are interested in this, I just show you a distribution of attendance per country so it could be perhaps useful to some of you. That is just for statistics, let us say. Now concerning the technical content, let me just say that I agreed with many of the things Dr. Kroll said. Anyway, I think that during this week we could at least answer some questions. For example, we know which was perhaps not obvious for all of us, that there is still a lot of exciting future for CFD. That is great because otherwise many of us would become unemployed in the near future. I feel also that we are all convinced that there is no way that CFD could replace wind tunnels in the future. They have to work together and they are very complementary to each other. Their complementary role should still be reassessed and used more intensively in the future. Then I have some conclusions concerning some work that could be done in the future. For example, I believe that that issue on grids will be very important and I think that there is no way to say that structured grids or unstructured grids will be better. They have to be used together. For example, I would like to remind you of that idea of hybrid grids and overlapping grids, and all these things. I have already been interested in the fact that you can have good error detectors and good error

estimators. It would be nice if we could generalize them to the transient error and to evaluate the error due to the time discretization. That would help a lot in unsteady flow calculations. I also appreciated the talks on DNS. For example, it was very clear that DNS could only be used in some small parts of the configuration, for example, and then the issue would be to develop multi-block techniques using DNS in some blocks, and some other models in other blocks. The communication between blocks obviously still has to be defined. This is an important issue for the future. Finally, I believe that to accelerate the calculations, which I believe is something very important, we should both use more efficient numerical techniques like implicit techniques and so on. Also, to have efficient computers. I liked a lot the talks on parallel computing, namely the talk by Prof. Knight. I must confess that many of us who don't use parallel computing are a little bit scared of using it. But I think we should do it anyway, or we will be out of business. I would however feel concerned by the portability issue. If you say it could become very portable, it would be nice to use it.

To close this meeting, I would like to thank all of the people who contributed to the success of this Conference. I will not thank each of them separately because this will be done by Christian Dujarric in a few minutes. I just would like to say that I am grateful to the authors who prepared good papers; in particular, I feel very satisfied by the fact that we received a copy of all of the papers now, which is not so usual in AGARD Conferences, so you can go back home with copies of all the papers. That is good in itself. I would also like to thank the Programme Committee members, the session chairmen and the technical ealuator, Dr. Kroll, who did a great job, in my opinion and also the Spanish organizers. They had planned everything including very good weather and they had a great party on Monday. There were very nice facilities. I would like to thank you for your attendance to this Symposium. I hope that you will go back home and remember this Conference as useful for your work. I hope that it will be very rewarding for your career, and wish you a good trip back home.

C. Dujarric, Chairman Fluid Dynamics Panel
Thank you Prof. Essers. Ladies and Gentlemen. We have now come to the end of our Symposium. I think that we have identified together promising research orientations. The scientific material will permit each of us to formulate recommendations for our respective organizations on the aspects of the use of numerical methods for aerodynamics which particularly merits our efforts for its development. The Fluid Dynamics Panel will use the results of this Conference as one of the elements for its contribution to Working Group Aerospace 2020. This Working Group will present to the highest authorities of NATO the recommendations concerning the technological efforts required to provide to the Alliance by 2020, radically improved military capacity in spite of the tight expected budgetary pressures.

A symposium regrouping all the panels of AGARD is planned in Paris in the Spring of 1997 to present the conclusions of this Working Group. This meeting will have in attendance military authorities, industry representatives and researchers. This will be for us an occasion to deliver our message, and I hope that many of you will participate.

This Symposium which has just finished, has been very well followed as Prof. Essers has mentioned. Inspite of some of the points made by Dr. Kroll, it was very fruitful. The Program Committee deserves our congratulations. We thank Prof. Essers, the Chairman of the Program Committee. We also thank the members of the Committee, Prof. Deconinck, Prof. Kind, Prof. Bonnet, M. Jacquotte, M. Lacau, Dr. Korner, Prof. Panaras, M. Borsi, Prof. Slooff, Dr. Ytrehus, Prof. Falcao, Dr. Corral, Prof. Jimenez, Prof. Kaynak, Dr. Poll, Prof. Cantwell and Dr. Lekoudis. We warmly thank all the authors and all of you who have helped us to have a lively discussion. We also thank the Technical Evaluator, Dr. Kroll, who has presented his point of view regarding our work. These comments will be attached to the publication of the Round Table Discussion. A remarkable job of organization was done to permit us to have our Symposium. I would like to thank on behalf of the Fluid Dynamics Panel, the Spanish authorities, in particular the National Delegates, for the invitation to hold this meeting in Seville. I remind you that the Minister of Defense for Spain and INTA have contributed to making our stay so agreeable by financing the organization of our Conference. We are very grateful. We thank in particular, Lt. General Mira Perez for the wonderful evening we had last Monday.

We especially thank our Local Coordinator, Prof. Javier Jimenez as well as Miss C. Gonzalez Hernandez, Spanish National Coordinator.

This Conference would not have been possible without the complicated logistics whose operation relies largely on good will. So we thank the interpreters who have succeeded in translating in spite of the very technical character of our remarks, considering especially the level of difficulty of doing so with certain speakers, perhaps myself included.

We thank the technicians for keeping the equipment functioning, the hostesses, as well as the people who welcomed us and helped in the smooth running of the Conference.

Lastly, we thank the Secretary of our Panel, Anne-Marie Rivault, who has just received the AGARD Personnel Medal for her devotion to the FDP and who participates for the last time in a Symposium before taking her well-deserved retirement.

We also thank the Panel's Executive, Mr. Jack Molloy for his very effective support in the preparation of this Conference.
Now I would like to present you with our program for 1996. We will have in the Spring a Symposium on the Characterization and Modifilation of Wakes from Lifting Vehicles. This will take place in Trondheim in Norway from the 20 to the 23th of May, 1996. In the Fall, if everything goes well, we will organize for the first time in the history of AGARD, a Symposium in Moscow. It demonstrates the recent opening up toward the countries of the old Soviet bloc. It will cover the Aerodynamics of Wind Tunnel Circuits and Their Components. The Russians have a great deal of experience in this

field and have promised to share this expertise. We begin a new level of cooperation which will be technically extremely fruitful for us. We will also have in 1996, two special courses at VKI, one on Advances in Cryogenic Wind Tunnel Technology, and the other on Aerothermodynamics and Propulsion Integration for Hypersonic Vehicles. You are all invited to participate in our future progams, and I hope to have the pleasure to meet you. Thank you for your attention.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's Reference	3. Further Reference	4. Security Classification of Document
	AGARD-CP-578	ISBN 92-836-0026-6	UNCLASSIFIED/ UNLIMITED

5. Originator Advisory Group for Aerospace Research and Development
North Atlantic Treaty Organization
7 rue Ancelle, 92200 Neuilly-sur-Seine, France

6. Title

Progress and Challenges in CFD Methods and Algorithms

7. Presented at/sponsored by

the 77th Fluid Dynamics Panel Symposium held in Seville, Spain, 2-5 October 1995

8. Author(s)/Editor(s)	9. Date
Multiple	April 1996

10. Author's/Editor's Address	11. Pages
Multiple	488

12. Distribution Statement There are no restrictions on the distribution of this document. Information about the availability of this and other AGARD unclassified publications is given on the back cover.

13. Keywords/Descriptors

Computational fluid dynamics
Design
Algorithms
Aerodynamics
Parallel processing
Parallel programming
Computer architecture
Unsteady flow
Turbulent flow
Aeroelasticity
Fluid dynamics
Chemical reactions
Computerized simulation
Computation
Grids (coordinates)

14. Abstract

The papers prepared for the AGARD Fluid Dynamics Panel (FDP) Symposium on “Progress and Challenges in CFD Methods and Algorithms”, which was held 2-5 October 1995 in Seville, Spain are contained in this Report. In addition, a Technical Evaluator's Report aimed at assessing the success of the Symposium in meeting its objectives, and an edited transcript of the General Discussion held at the end of the Symposium are also included.

Papers presented during nine sessions addressed the following subjects:

— parallel computing;
— advanced spatial descretization techniques;
— unstructured, hybrid and overlapping grids;
— adaptive meshes;
— fast implicit and iterative solvers;
— large eddy and direct numerical simulations of turbulent flows;
— chemically reacting flows;
— unsteady aerodynamics.

AGARD

NATO OTAN

7 RUE ANCELLE • 92200 NEUILLY-SUR-SEINE

FRANCE

Télécopie (1)47.38.57.99 • Télex 610 176

DIFFUSION DES PUBLICATIONS

AGARD NON CLASSIFIEES

Aucun stock de publications n'a existé à AGARD. A partir de 1993, AGARD détiendra un stock limité des publications associées aux cycles de conférences et cours spéciaux ainsi que les AGARDographies et les rapports des groupes de travail, organisés et publiés à partir de 1993 inclus. Les demandes de renseignements doivent être adressées à AGARD par lettre ou par fax à l'adresse indiquée ci-dessus. *Veuillez ne pas téléphoner*. La diffusion initiale de toutes les publications de l'AGARD est effectuée auprès des pays membres de l'OTAN par l'intermédiaire des centres de distribution nationaux indiqués ci-dessous. Des exemplaires supplémentaires peuvent parfois être obtenus auprès de ces centres (à l'exception des Etats-Unis). Si vous souhaitez recevoir toutes les publications de l'AGARD, ou simplement celles qui concernent certains Panels, vous pouvez demander à être inclu sur la liste d'envoi de l'un de ces centres. Les publications de l'AGARD sont en vente auprès des agences indiquées ci-dessous, sous forme de photocopie ou de microfiche.

CENTRES DE DIFFUSION NATIONAUX

ALLEMAGNE
Fachinformationszentrum Karlsruhe
D-76344 Eggenstein-Leopoldshafen 2

BELGIQUE
Coordonnateur AGARD-VSL
Etat-major de la Force aérienne
Quartier Reine Elisabeth
Rue d'Evere, 1140 Bruxelles

CANADA
Directeur, Services d'information scientifique
Ministère de la Défense nationale
Ottawa, Ontario K1A 0K2

DANEMARK
Danish Defence Research Establishment
Ryvangs Allé 1
P.O. Box 2715
DK-2100 Copenhagen Ø

ESPAGNE
INTA (AGARD Publications)
Carretera de Torrejón a Ajalvir, Pk.4
28850 Torrejón de Ardoz - Madrid

ETATS-UNIS
NASA Headquarters
Code JOB-1
Washington, D.C. 20546

FRANCE
O.N.E.R.A. (Direction)
29, Avenue de la Division Leclerc
92322 Châtillon Cedex

GRECE
Hellenic Air Force
Air War College
Scientific and Technical Library
Dekelia Air Force Base
Dekelia, Athens TGA 1010

ISLANDE
Director of Aviation
c/o Flugrad
Reykjavik

ITALIE
Aeronautica Militare
Ufficio del Delegato Nazionale all'AGARD
Aeroporto Pratica di Mare
00040 Pomezia (Roma)

LUXEMBOURG
Voir Belgique

NORVEGE
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25
N-2007 Kjeller

PAYS-BAS
Netherlands Delegation to AGARD
National Aerospace Laboratory NLR
P.O. Box 90502
1006 BM Amsterdam

PORTUGAL
Estado Maior da Força Aérea
SDFA - Centro de Documentação
Alfragide
2700 Amadora

ROYAUME-UNI
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

TURQUIE
Millî Savunma Başkanliği (MSB)
ARGE Dairesi Başkanliği (MSB)
06650 Bakanliklar-Ankara

Le centre de distribution national des Etats-Unis ne détient PAS de stocks des publications de l'AGARD.

D'éventuelles demandes de photocopies doivent être formulées directement auprès du NASA Center for AeroSpace Information (CASI) à l'adresse ci-dessous. Toute notification de changement d'adresse doit être fait également auprès de CASI.

AGENCES DE VENTE

NASA Center for
AeroSpace Information (CASI)
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
Etats-Unis

ESA/Information Retrieval Service
European Space Agency
10, rue Mario Nikis
75015 Paris
France

The British Library
Document Supply Division
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
Royaume-Uni

Les demandes de microfiches ou de photocopies de documents AGARD (y compris les demandes faites auprès du CASI) doivent comporter la dénomination AGARD, ainsi que le numéro de série d'AGARD (par exemple AGARD-AG-315). Des informations analogues, telles que le titre et la date de publication sont souhaitables. Veuiller noter qu'il y a lieu de spécifier AGARD-R-nnn et AGARD-AR-nnn lors de la commande des rapports AGARD et des rapports consultatifs AGARD respectivement. Des références bibliographiques complètes ainsi que des résumés des publications AGARD figurent dans les journaux suivants:

Scientific and Technical Aerospace Reports (STAR)
publié par la NASA Scientific and Technical
Information Division
NASA Headquarters (JTT)
Washington D.C. 20546
Etats-Unis

Government Reports Announcements and Index (GRA&I)
publié par le National Technical Information Service
Springfield
Virginia 22161
Etats-Unis
(accessible également en mode interactif dans la base de
données bibliographiques en ligne du NTIS, et sur CD-ROM)

Imprimé par le Groupe Communication Canada
45, boul. Sacré-Cœur, Hull (Québec), Canada K1A 0S7

AGARD

NATO OTAN

7 RUE ANCELLE • 92200 NEUILLY-SUR-SEINE

FRANCE

Telefax (1)47.38.57.99 • Telex 610 176

DISTRIBUTION OF UNCLASSIFIED AGARD PUBLICATIONS

AGARD holds limited quantities of the publications that accompanied Lecture Series and Special Courses held in 1993 or later, and of AGARDographs and Working Group reports published from 1993 onward. For details, write or send a telefax to the address given above. *Please do not telephone.*

AGARD does not hold stocks of publications that accompanied earlier Lecture Series or Courses or of any other publications. Initial distribution of all AGARD publications is made to NATO nations through the National Distribution Centres listed below. Further copies are sometimes available from these centres (except in the United States). If you have a need to receive all AGARD publications, or just those relating to one or more specific AGARD Panels, they may be willing to include you (or your organisation) on their distribution list. AGARD publications may be purchased from the Sales Agencies listed below, in photocopy or microfiche form.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Coordonnateur AGARD — VSL
Etat-major de la Force aérienne
Quartier Reine Elisabeth
Rue d'Evere, 1140 Bruxelles

CANADA
Director Scientific Information Services
Dept of National Defence
Ottawa, Ontario K1A 0K2

DENMARK
Danish Defence Research Establishment
Ryvangs Allé 1
P.O. Box 2715
DK-2100 Copenhagen Ø

FRANCE
O.N.E.R.A. (Direction)
29 Avenue de la Division Leclerc
92322 Châtillon Cedex

GERMANY
Fachinformationszentrum Karlsruhe
D-76344 Eggenstein-Leopoldshafen 2

GREECE
Hellenic Air Force
Air War College
Scientific and Technical Library
Dekelia Air Force Base
Dekelia, Athens TGA 1010

ICELAND
Director of Aviation
c/o Flugrad
Reykjavik

ITALY
Aeronautica Militare
Ufficio del Delegato Nazionale all'AGARD
Aeroporto Pratica di Mare
00040 Pomezia (Roma)

LUXEMBOURG
See Belgium

NETHERLANDS
Netherlands Delegation to AGARD
National Aerospace Laboratory, NLR
P.O. Box 90502
1006 BM Amsterdam

NORWAY
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25
N-2007 Kjeller

PORTUGAL
Estado Maior da Força Aérea
SDFA - Centro de Documentação
Alfragide
2700 Amadora

SPAIN
INTA (AGARD Publications)
Carretera de Torrejón a Ajalvir, Pk.4
28850 Torrejón de Ardoz - Madrid

TURKEY
Millî Savunma Başkanliği (MSB)
ARGE Dairesi Başkanliği (MSB)
06650 Bakanliklar-Ankara

UNITED KINGDOM
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

UNITED STATES
NASA Headquarters
Code JOB-1
Washington, D.C. 20546

The United States National Distribution Centre does NOT hold stocks of AGARD publications.
Applications for copies should be made direct to the NASA Center for AeroSpace Information (CASI) at the address below.
Change of address requests should also go to CASI.

SALES AGENCIES

NASA Center for
AeroSpace Information (CASI)
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
United States

ESA/Information Retrieval Service
European Space Agency
10, rue Mario Nikis
75015 Paris
France

The British Library
Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
United Kingdom

Requests for microfiches or photocopies of AGARD documents (including requests to CASI) should include the word 'AGARD' and the AGARD serial number (for example AGARD-AG-315). Collateral information such as title and publication date is desirable. Note that AGARD Reports and Advisory Reports should be specified as AGARD-R-nnn and AGARD-AR-nnn, respectively. Full bibliographical references and abstracts of AGARD publications are given in the following journals:

Scientific and Technical Aerospace Reports (STAR)
published by NASA Scientific and Technical
Information Division
NASA Headquarters (JTT)
Washington D.C. 20546
United States

Government Reports Announcements and Index (GRA&I)
published by the National Technical Information Service
Springfield
Virginia 22161
United States
(also available online in the NTIS Bibliographic
Database or on CD-ROM)

Printed by Canada Communication Group
45 Sacré-Cœur Blvd., Hull (Québec), Canada K1A 0S7

ISBN 92-836-0026-6